CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 8415 from hexadecimal to base 2 *1000 0100 0001 0101*

   b. (2 pts) Convert 57 from octal to base 2 *101 111*

   c. (2 pts) Convert 0100 1001 0100 1010 from base 2 to hexadecimal *494a*

   d. (2 pts) Convert 75 from octal to base 2 *111 101*

   e. (2 pts) Convert 1110 0101 from binary to decimal *229*

   f. (2 pts) Convert 0101 1010 from base 2 to decimal *90*

   g. (2 pts) Convert 6086 from hexadecimal to base 2 *0110 0000 1000 0110*

   h. (2 pts) Convert 129 from decimal to base 2 *1000 0001*

   i. (2 pts) Convert 2fc6 from base 16 to base 2 *0010 1111 1100 0110*

   j. (2 pts) Convert 111 011 000 from base 2 to base 8 *730*

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  Node x;
  double y;
  int z;
  char a;
  Node *b;
  double *c;
  int *d;
  char *e;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) &c double **

b. (2 pts) argv[0] char *

c. (2 pts) *e char

d. (2 pts) b->next->next Node *

e. (2 pts) z int

f. (2 pts) b->next Node *

g. (2 pts) argc int

h. (2 pts) b->data int

i. (2 pts) argv[1][2] char

j. (2 pts) d int *

k. (2 pts) &a char *

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt lime lemon
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of argv[2][3]? o

c. (2 pts) What is the value of argv[1][0]? l

d. (2 pts) What is the value of argv[0][5]? I

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 181 from base 10 to binary 1011 0101

   b. (2 pts) Convert 43 from base 8 to binary 100 011

   c. (2 pts) Convert 127 from base 10 to base 2 0111 1111

   d. (2 pts) Convert 010 100 010 from binary to base 8 242

   e. (2 pts) Convert 59 from decimal to binary 0011 1011

   f. (2 pts) Convert 1000 1100 1001 0011 from base 2 to base 16 8c93

   g. (2 pts) Convert 62b4 from base 16 to binary 0110 0010 1011 0100

   h. (2 pts) Convert 234 from base 10 to base 2 1110 1010

   i. (2 pts) Convert 209 from decimal to binary 1101 0001

   j. (2 pts) Convert 0001 1000 from base 2 to decimal 24

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double g;
  int h;
  Node p;
  char q;
  double *r;
  int *s;
  Node *t;
  char *w;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) argv[1][2] *char*

b. (2 pts) argv[0] *char **

c. (2 pts) q *char*

d. (2 pts) *t *Node*

e. (2 pts) &h *int **

f. (2 pts) argc *int*

g. (2 pts) t->data *int*

h. (2 pts) t->next->next *Node **

i. (2 pts) t *Node **

j. (2 pts) t->next *Node **

k. (2 pts) &t *Node ***

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt kiwi lemon date guava
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of argv[1][2]? w

c. (2 pts) What is the value of argv[0][4]? n

d. (2 pts) What is the value of argv[2][0]? l

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 1111 1010 0101 0110 from base 2 to hexadecimal fa56

   b. (2 pts) Convert 0100 1001 from base 2 to base 10 73

   c. (2 pts) Convert 109 from decimal to base 2 0110 1101

   d. (2 pts) Convert 110 111 010 from binary to octal 672

   e. (2 pts) Convert 0001 1110 1110 1000 from binary to base 16 1ee8

   f. (2 pts) Convert 2670 from hexadecimal to base 2 0010 0110 0111 0000

   g. (2 pts) Convert ca5 from hexadecimal to base 2 1100 1010 0101

   h. (2 pts) Convert 34 from base 8 to base 2 011 100

   i. (2 pts) Convert 9b98 from base 16 to base 2 1001 1011 1001 1000

   j. (2 pts) Convert 1011 0100 from binary to base 10 180

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  Node d;
  int e;
  double f;
  char g;
  Node *h;
  int *p;
  double *q;
  char *r;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) &p int **

b. (2 pts) p int *

c. (2 pts) *h Node

d. (2 pts) &e int *

e. (2 pts) argc int

f. (2 pts) h->next->next Node *

g. (2 pts) argv[0] char *

h. (2 pts) e int

i. (2 pts) h->data int

j. (2 pts) h->next Node *

k. (2 pts) argv[1][2] char

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt date apple kiwi
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of argv[0][5]? I

c. (2 pts) What is the value of argv[1][3]? e

d. (2 pts) What is the value of argv[2][3]? l

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 24 from base 8 to binary 010 100

   b. (2 pts) Convert 111 011 001 from base 2 to octal 731

   c. (2 pts) Convert 0001 0010 from base 2 to decimal 18

   d. (2 pts) Convert 152 from base 10 to binary 1001 1000

   e. (2 pts) Convert 100 011 110 from binary to octal 436

   f. (2 pts) Convert 001 001 111 from binary to base 8 117

   g. (2 pts) Convert 0101 1110 0101 1000 from base 2 to hexadecimal 5e58

   h. (2 pts) Convert 6 from base 8 to base 2 110

   i. (2 pts) Convert 8e0c from base 16 to binary 1000 1110 0000 1100

   j. (2 pts) Convert 101 111 111 from binary to octal 577

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  Node w;
  int x;
  double y;
  char z;
  Node *a;
  int *b;
  double *c;
  char *d;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) argv[0] *char **

b. (2 pts) &w *Node **

c. (2 pts) a->data *int*

d. (2 pts) argc *int*

e. (2 pts) a->next->next *Node **

f. (2 pts) a->next *Node **

g. (2 pts) w *Node*

h. (2 pts) argv[1][2] *char*

i. (2 pts) &d *char ***

j. (2 pts) *d *char*

k. (2 pts) d *char **

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt cherry banana date
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of argv[2][3]? a

c. (2 pts) What is the value of argv[1][3]? r

d. (2 pts) What is the value of argv[0][4]? n

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 1001 0111 from base 2 to base 10 151

   b. (2 pts) Convert 1010 0110 from binary to decimal 166

   c. (2 pts) Convert 0000 from binary to decimal 0

   d. (2 pts) Convert 36 from base 10 to base 2 0010 0100

   e. (2 pts) Convert 114 from decimal to binary 0111 0010

   f. (2 pts) Convert 110 000 011 from binary to octal 603

   g. (2 pts) Convert 010 000 from binary to octal 20

   h. (2 pts) Convert 160 from decimal to base 2 1010 0000

   i. (2 pts) Convert 88 from decimal to base 2 0101 1000

   j. (2 pts) Convert 010 110 111 from binary to octal 267

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double r;
  Node s;
  int t;
  char w;
  double *x;
  Node *y;
  int *z;
  char *a;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) w *char*

b. (2 pts) y->next *Node **

c. (2 pts) argc *int*

d. (2 pts) argv[0] *char **

e. (2 pts) z *int **

f. (2 pts) y->next->next *Node **

g. (2 pts) argv[1][2] *char*

h. (2 pts) y->data *int*

i. (2 pts) &w *char **

j. (2 pts) &z *int ***

k. (2 pts) *y *Node*

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt guava lime
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of argv[1][0]? g

c. (2 pts) What is the value of argv[2][1]? i

d. (2 pts) What is the value of argv[0][1]? /

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 110 010 001 from binary to base 8 621

   b. (2 pts) Convert 120 from decimal to binary 0111 1000

   c. (2 pts) Convert 0011 0111 0010 1011 from binary to hexadecimal 372b

   d. (2 pts) Convert 8101 from base 16 to binary 1000 0001 0000 0001

   e. (2 pts) Convert c8ca from hexadecimal to binary 1100 1000 1100 1010

   f. (2 pts) Convert 111 101 000 from base 2 to octal 750

   g. (2 pts) Convert a77 from base 16 to binary 1010 0111 0111

   h. (2 pts) Convert 9 from decimal to base 2 1001

   i. (2 pts) Convert f9dd from base 16 to binary 1111 1001 1101 1101

   j. (2 pts) Convert 100 010 000 from binary to base 8 420

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int r;
  double s;
  Node t;
  char w;
  int *x;
  double *y;
  Node *z;
  char *a;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) z->next->next *Node **

b. (2 pts) argv[1][2] *char*

c. (2 pts) z->data *int*

d. (2 pts) *x *int*

e. (2 pts) &a *char ***

f. (2 pts) &t *Node **

g. (2 pts) y *double **

h. (2 pts) z->next *Node **

i. (2 pts) argv[0] *char **

j. (2 pts) r *int*

k. (2 pts) argc *int*

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt kiwi fig guava apple
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of argv[0][5]? I

c. (2 pts) What is the value of argv[1][2]? w

d. (2 pts) What is the value of argv[2][2]? g

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 13 from base 10 to binary 1101

   b. (2 pts) Convert 328e from base 16 to base 2 0011 0010 1000 1110

   c. (2 pts) Convert 250b from base 16 to binary 0010 0101 0000 1011

   d. (2 pts) Convert ceb from base 16 to base 2 1100 1110 1011

   e. (2 pts) Convert 53 from octal to base 2 101 011

   f. (2 pts) Convert 43 from octal to base 2 100 011

   g. (2 pts) Convert b467 from hexadecimal to binary 1011 0100 0110 0111

   h. (2 pts) Convert 44 from base 8 to base 2 100 100

   i. (2 pts) Convert 195 from decimal to base 2 1100 0011

   j. (2 pts) Convert 001 000 111 from base 2 to base 8 107

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double z;
  int a;
  Node b;
  char c;
  double *d;
  int *e;
  Node *f;
  char *g;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) f->next *Node **

b. (2 pts) d *double **

c. (2 pts) b *Node*

d. (2 pts) *f *Node*

e. (2 pts) argv[0] *char **

f. (2 pts) argv[1][2] *char*

g. (2 pts) &e *int ***

h. (2 pts) f->next->next *Node **

i. (2 pts) argc *int*

j. (2 pts) &a *int **

k. (2 pts) f->data *int*

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt date cherry mango fig
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of argv[1][1]? a

c. (2 pts) What is the value of argv[0][4]? n

d. (2 pts) What is the value of argv[2][3]? r

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 202 from decimal to base 2 1100 1010

   b. (2 pts) Convert 9041 from base 16 to binary 1001 0000 0100 0001

   c. (2 pts) Convert 1110 from base 2 to base 10 14

   d. (2 pts) Convert 0100 1011 0011 1000 from binary to hexadecimal 4b38

   e. (2 pts) Convert 54 from base 8 to binary 101 100

   f. (2 pts) Convert 0101 0011 1001 0111 from binary to hexadecimal 5397

   g. (2 pts) Convert 1100 1000 0000 0111 from binary to hexadecimal c807

   h. (2 pts) Convert 010 000 011 from binary to octal 203

   i. (2 pts) Convert 0111 0011 1000 0100 from base 2 to base 16 7384

   j. (2 pts) Convert 1011 0010 from base 2 to decimal 178

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int x;
  Node y;
  double z;
  char a;
  int *b;
  Node *c;
  double *d;
  char *e;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) argv[1][2] *char*

b. (2 pts) &e *char* **

c. (2 pts) &y *Node* *

d. (2 pts) *d *double*

e. (2 pts) c->data *int*

f. (2 pts) c->next *Node* *

g. (2 pts) y *Node*

h. (2 pts) argc *int*

i. (2 pts) d *double* *

j. (2 pts) c->next->next *Node* *

k. (2 pts) argv[0] *char* *

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt kiwi lemon cherry
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of argv[2][4]? n

c. (2 pts) What is the value of argv[0][5]? I

d. (2 pts) What is the value of argv[1][0]? k

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 3 from octal to base 2 011

   b. (2 pts) Convert 0100 1010 0110 1001 from base 2 to base 16 4a69

   c. (2 pts) Convert 77 from octal to base 2 111 111

   d. (2 pts) Convert 65 from octal to base 2 110 101

   e. (2 pts) Convert 1001 0110 from base 2 to base 10 150

   f. (2 pts) Convert 1110 1101 0111 0100 from binary to hexadecimal ed74

   g. (2 pts) Convert 0111 0001 1111 0111 from base 2 to base 16 71f7

   h. (2 pts) Convert 201 from base 10 to binary 1100 1001

   i. (2 pts) Convert 17 from octal to binary 001 111

   j. (2 pts) Convert 0100 1110 from binary to decimal 78

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int s;
  Node t;
  double w;
  char x;
  int *y;
  Node *z;
  double *a;
  char *b;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) z Node *

b. (2 pts) &z Node **

c. (2 pts) argc int

d. (2 pts) z->data int

e. (2 pts) z->next->next Node *

f. (2 pts) argv[1][2] char

g. (2 pts) *b char

h. (2 pts) &s int *

i. (2 pts) z->next Node *

j. (2 pts) argv[0] char *

k. (2 pts) x char

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt fig grape guava
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of argv[2][1]? r

c. (2 pts) What is the value of argv[1][0]? f

d. (2 pts) What is the value of argv[0][1]? /

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 4088 from base 16 to binary 0100 0000 1000 1000

   b. (2 pts) Convert 110 111 from binary to octal 67

   c. (2 pts) Convert 32a0 from hexadecimal to binary 0011 0010 1010 0000

   d. (2 pts) Convert 001 100 110 from binary to octal 146

   e. (2 pts) Convert 111 011 001 from base 2 to base 8 731

   f. (2 pts) Convert 111 111 from base 2 to base 8 77

   g. (2 pts) Convert 0111 0100 0010 0101 from binary to hexadecimal 7425

   h. (2 pts) Convert 3246 from base 16 to binary 0011 0010 0100 0110

   i. (2 pts) Convert 67 from base 8 to base 2 110 111

   j. (2 pts) Convert 0111 1010 from binary to decimal 122

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double e;
  Node f;
  int g;
  char h;
  double *p;
  Node *q;
  int *r;
  char *s;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) p double *

b. (2 pts) argc int

c. (2 pts) argv[1][2] char

d. (2 pts) q->next Node *

e. (2 pts) &g int *

f. (2 pts) argv[0] char *

g. (2 pts) e double

h. (2 pts) *s char

i. (2 pts) q->data int

j. (2 pts) &r int **

k. (2 pts) q->next->next Node *

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt lime banana
```

a. (2 pts) What is the value of `argc` in this case? *3*

b. (2 pts) What is the value of argv[1][2]? *m*

c. (2 pts) What is the value of argv[0][6]? *t*

d. (2 pts) What is the value of argv[2][2]? *n*

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 100 001 010 from base 2 to base 8 412

   b. (2 pts) Convert 110 101 100 from base 2 to base 8 654

   c. (2 pts) Convert 2081 from hexadecimal to binary 0010 0000 1000 0001

   d. (2 pts) Convert 101 111 110 from binary to base 8 576

   e. (2 pts) Convert 207 from base 10 to binary 1100 1111

   f. (2 pts) Convert 56 from base 8 to base 2 101 110

   g. (2 pts) Convert 0001 1110 0001 0110 from base 2 to hexadecimal 1e16

   h. (2 pts) Convert 56 from octal to binary 101 110

   i. (2 pts) Convert 52 from octal to binary 101 010

   j. (2 pts) Convert 0001 0110 from base 2 to decimal 22

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double a;
  int b;
  Node c;
  char d;
  double *e;
  int *f;
  Node *g;
  char *h;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) argv[1][2] *char*

b. (2 pts) g->data *int*

c. (2 pts) argv[0] *char **

d. (2 pts) &f *int ***

e. (2 pts) g->next *Node **

f. (2 pts) c *Node*

g. (2 pts) *e *double*

h. (2 pts) h *char **

i. (2 pts) g->next->next *Node **

j. (2 pts) argc *int*

k. (2 pts) &b *int **

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt date cherry fig apple
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of argv[0][3]? u

c. (2 pts) What is the value of argv[1][0]? d

d. (2 pts) What is the value of argv[2][0]? c

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 110 111 011 from base 2 to base 8 673

   b. (2 pts) Convert 36 from base 8 to binary 011 110

   c. (2 pts) Convert 1100 0101 1110 0010 from binary to base 16 c5e2

   d. (2 pts) Convert 0111 1010 1110 0001 from base 2 to base 16 7ae1

   e. (2 pts) Convert 010 000 000 from base 2 to base 8 200

   f. (2 pts) Convert bb0b from base 16 to base 2 1011 1011 0000 1011

   g. (2 pts) Convert 011 011 111 from base 2 to octal 337

   h. (2 pts) Convert 96 from decimal to binary 0110 0000

   i. (2 pts) Convert 9be1 from base 16 to base 2 1001 1011 1110 0001

   j. (2 pts) Convert 001 000 011 from binary to base 8 103

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  Node r;
  int s;
  double t;
  char w;
  Node *x;
  int *y;
  double *z;
  char *a;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) argc *int*

b. (2 pts) &s *int **

c. (2 pts) argv[1][2] *char*

d. (2 pts) y *int **

e. (2 pts) &a *char ***

f. (2 pts) x->next *Node **

g. (2 pts) argv[0] *char **

h. (2 pts) t *double*

i. (2 pts) *x *Node*

j. (2 pts) x->next->next *Node **

k. (2 pts) x->data *int*

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt cherry lemon banana
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of argv[2][1]? e

c. (2 pts) What is the value of argv[0][1]? /

d. (2 pts) What is the value of argv[1][2]? e

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 34 from base 10 to base 2 *0010 0010*

   b. (2 pts) Convert 51 from base 10 to binary *0011 0011*

   c. (2 pts) Convert 179 from decimal to binary *1011 0011*

   d. (2 pts) Convert 6 from base 10 to binary *0110*

   e. (2 pts) Convert 35 from base 10 to base 2 *0010 0011*

   f. (2 pts) Convert 84 from decimal to binary *0101 0100*

   g. (2 pts) Convert 110 011 from binary to octal *63*

   h. (2 pts) Convert e819 from hexadecimal to binary *1110 1000 0001 1001*

   i. (2 pts) Convert 0110 0101 1111 1000 from binary to hexadecimal *65f8*

   j. (2 pts) Convert 101 111 011 from base 2 to base 8 *573*

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  Node h;
  int p;
  double q;
  char r;
  Node *s;
  int *t;
  double *w;
  char *x;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) &w double **

b. (2 pts) s->next->next Node *

c. (2 pts) h Node

d. (2 pts) s->next Node *

e. (2 pts) argv[0] char *

f. (2 pts) argv[1][2] char

g. (2 pts) s->data int

h. (2 pts) *t int

i. (2 pts) argc int

j. (2 pts) s Node *

k. (2 pts) &h Node *

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt date guava lemon
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of argv[1][1]? a

c. (2 pts) What is the value of argv[2][1]? u

d. (2 pts) What is the value of argv[0][4]? n

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 0101 0100 from binary to decimal 84

   b. (2 pts) Convert 0100 from binary to decimal 4

   c. (2 pts) Convert 111 010 100 from binary to base 8 724

   d. (2 pts) Convert 630d from hexadecimal to binary 0110 0011 0000 1101

   e. (2 pts) Convert 79ba from base 16 to binary 0111 1001 1011 1010

   f. (2 pts) Convert 8751 from base 16 to binary 1000 0111 0101 0001

   g. (2 pts) Convert 110 111 from binary to base 8 67

   h. (2 pts) Convert 24 from base 8 to binary 010 100

   i. (2 pts) Convert 0111 1011 0011 from base 2 to base 16 7b3

   j. (2 pts) Convert 1110 1010 from base 2 to decimal 234

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  Node a;
  double b;
  int c;
  char d;
  Node *e;
  double *f;
  int *g;
  char *h;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) e *Node **

b. (2 pts) argc *int*

c. (2 pts) *f *double*

d. (2 pts) b *double*

e. (2 pts) e->next->next *Node **

f. (2 pts) &d *char **

g. (2 pts) e->data *int*

h. (2 pts) argv[0] *char **

i. (2 pts) argv[1][2] *char*

j. (2 pts) e->next *Node **

k. (2 pts) &h *char ***

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt mango guava
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of argv[1][2]? n

c. (2 pts) What is the value of argv[0][4]? n

d. (2 pts) What is the value of argv[2][1]? u

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 98e4 from base 16 to binary 1001 1000 1110 0100

   b. (2 pts) Convert 190 from decimal to base 2 1011 1110

   c. (2 pts) Convert 1101 1000 0000 0010 from base 2 to base 16 d802

   d. (2 pts) Convert eef7 from base 16 to binary 1110 1110 1111 0111

   e. (2 pts) Convert 0101 1101 0000 0101 from base 2 to hexadecimal 5d05

   f. (2 pts) Convert 0010 0001 0010 1110 from base 2 to base 16 212e

   g. (2 pts) Convert 110 001 011 from binary to base 8 613

   h. (2 pts) Convert 66 from base 8 to binary 110 110

   i. (2 pts) Convert 64 from octal to binary 110 100

   j. (2 pts) Convert 100 001 100 from base 2 to base 8 414

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double x;
  int y;
  Node z;
  char a;
  double *b;
  int *c;
  Node *d;
  char *e;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) d *Node **

b. (2 pts) &z *Node **

c. (2 pts) d->next->next *Node **

d. (2 pts) a *char*

e. (2 pts) d->next *Node **

f. (2 pts) argv[1][2] *char*

g. (2 pts) *d *Node*

h. (2 pts) &c *int ***

i. (2 pts) argc *int*

j. (2 pts) d->data *int*

k. (2 pts) argv[0] *char **

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt banana fig lemon date
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of argv[0][1]? /

c. (2 pts) What is the value of argv[1][3]? a

d. (2 pts) What is the value of argv[2][1]? i

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 1111 0001 0100 0001 from binary to hexadecimal f141

   b. (2 pts) Convert 011 000 011 from binary to base 8 303

   c. (2 pts) Convert 37 from octal to binary 011 111

   d. (2 pts) Convert 1010 1010 from base 2 to base 10 170

   e. (2 pts) Convert 1100 1101 from base 2 to base 10 205

   f. (2 pts) Convert 001 000 100 from base 2 to base 8 104

   g. (2 pts) Convert 0001 0111 from base 2 to base 10 23

   h. (2 pts) Convert 127 from base 10 to binary 0111 1111

   i. (2 pts) Convert 61 from base 8 to base 2 110 001

   j. (2 pts) Convert 44 from base 8 to binary 100 100

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  Node h;
  double p;
  int q;
  char r;
  Node *s;
  double *t;
  int *w;
  char *x;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) t *double **

b. (2 pts) s->data *int*

c. (2 pts) argv[0] *char **

d. (2 pts) argv[1][2] *char*

e. (2 pts) *w *int*

f. (2 pts) s->next *Node **

g. (2 pts) argc *int*

h. (2 pts) &q *int **

i. (2 pts) s->next->next *Node **

j. (2 pts) &s *Node ***

k. (2 pts) r *char*

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt lemon lime
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of argv[0][3]? u

c. (2 pts) What is the value of argv[2][1]? i

d. (2 pts) What is the value of argv[1][1]? e

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 001 101 100 from base 2 to base 8 *154*

   b. (2 pts) Convert 1bb0 from base 16 to binary *0001 1011 1011 0000*

   c. (2 pts) Convert 6b44 from base 16 to binary *0110 1011 0100 0100*

   d. (2 pts) Convert 15 from base 8 to binary *001 101*

   e. (2 pts) Convert 176 from base 10 to binary *1011 0000*

   f. (2 pts) Convert 57 from octal to base 2 *101 111*

   g. (2 pts) Convert c17c from hexadecimal to base 2 *1100 0001 0111 1100*

   h. (2 pts) Convert 1 from octal to binary *001*

   i. (2 pts) Convert 8e55 from hexadecimal to binary *1000 1110 0101 0101*

   j. (2 pts) Convert 13 from octal to binary *001 011*

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double d;
  Node e;
  int f;
  char g;
  double *h;
  Node *p;
  int *q;
  char *r;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) &q *int* **

b. (2 pts) e *Node*

c. (2 pts) p->data *int*

d. (2 pts) p->next->next *Node* *

e. (2 pts) &e *Node* *

f. (2 pts) argv[0] *char* *

g. (2 pts) h *double* *

h. (2 pts) argv[1][2] *char*

i. (2 pts) p->next *Node* *

j. (2 pts) argc *int*

k. (2 pts) *r *char*

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt fig lemon
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of argv[1][1]? i

c. (2 pts) What is the value of argv[2][2]? m

d. (2 pts) What is the value of argv[0][1]? /

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 31 from octal to base 2 011 001

   b. (2 pts) Convert ed35 from hexadecimal to base 2 1110 1101 0011 0101

   c. (2 pts) Convert 101 000 011 from base 2 to octal 503

   d. (2 pts) Convert 146 from base 10 to base 2 1001 0010

   e. (2 pts) Convert 6e7 from base 16 to binary 0110 1110 0111

   f. (2 pts) Convert 1110 1110 from binary to decimal 238

   g. (2 pts) Convert 1100 0011 from base 2 to decimal 195

   h. (2 pts) Convert 33 from base 8 to binary 011 011

   i. (2 pts) Convert 300f from base 16 to base 2 0011 0000 0000 1111

   j. (2 pts) Convert 010 110 011 from binary to base 8 263

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int e;
  double f;
  Node g;
  char h;
  int *p;
  double *q;
  Node *r;
  char *s;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) r->data *int*

b. (2 pts) g *Node*

c. (2 pts) r->next *Node **

d. (2 pts) *s *char*

e. (2 pts) argv[1][2] *char*

f. (2 pts) r->next->next *Node **

g. (2 pts) argv[0] *char **

h. (2 pts) s *char **

i. (2 pts) &p *int ***

j. (2 pts) &h *char **

k. (2 pts) argc *int*

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt apple date guava grape
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of argv[2][1]? a

c. (2 pts) What is the value of argv[1][1]? p

d. (2 pts) What is the value of argv[0][2]? r

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 1010 1100 from binary to decimal 172

   b. (2 pts) Convert 51 from octal to binary 101 001

   c. (2 pts) Convert 1000 1111 from base 2 to decimal 143

   d. (2 pts) Convert 0001 1110 from base 2 to decimal 30

   e. (2 pts) Convert 72 from base 8 to base 2 111 010

   f. (2 pts) Convert 100 010 001 from base 2 to octal 421

   g. (2 pts) Convert 0110 1101 from binary to decimal 109

   h. (2 pts) Convert 111 101 110 from base 2 to octal 756

   i. (2 pts) Convert 1111 1010 0010 0110 from binary to hexadecimal fa26

   j. (2 pts) Convert 111 101 011 from binary to base 8 753

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int r;
  Node s;
  double t;
  char w;
  int *x;
  Node *y;
  double *z;
  char *a;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) argv[1][2] *char*

b. (2 pts) &z *double* **

c. (2 pts) argc *int*

d. (2 pts) y->next->next *Node* *

e. (2 pts) y->data *int*

f. (2 pts) s *Node*

g. (2 pts) z *double* *

h. (2 pts) argv[0] *char* *

i. (2 pts) &t *double* *

j. (2 pts) *x *int*

k. (2 pts) y->next *Node* *

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt kiwi lime fig
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of argv[2][1]? i

c. (2 pts) What is the value of argv[0][4]? n

d. (2 pts) What is the value of argv[1][2]? w

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 0100 from binary to decimal 4

   b. (2 pts) Convert 0100 1010 0010 1010 from binary to base 16 4a2a

   c. (2 pts) Convert 52 from base 10 to base 2 0011 0100

   d. (2 pts) Convert 1101 1010 0011 0011 from base 2 to hexadecimal da33

   e. (2 pts) Convert 0101 1010 from base 2 to decimal 90

   f. (2 pts) Convert 1000 1001 1111 0010 from base 2 to base 16 89f2

   g. (2 pts) Convert bf4e from base 16 to binary 1011 1111 0100 1110

   h. (2 pts) Convert 157 from base 10 to base 2 1001 1101

   i. (2 pts) Convert 1110 1100 1001 1010 from binary to hexadecimal ec9a

   j. (2 pts) Convert 0 from octal to binary 000

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int d;
  Node e;
  double f;
  char g;
  int *h;
  Node *p;
  double *q;
  char *r;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) p->next->next *Node **

b. (2 pts) p->data *int*

c. (2 pts) p *Node **

d. (2 pts) argv[0] *char **

e. (2 pts) d *int*

f. (2 pts) &p *Node ***

g. (2 pts) argv[1][2] *char*

h. (2 pts) argc *int*

i. (2 pts) *h *int*

j. (2 pts) p->next *Node **

k. (2 pts) &f *double **

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt banana lime guava
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of argv[0][1]? /

c. (2 pts) What is the value of argv[2][2]? m

d. (2 pts) What is the value of argv[1][1]? a

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 0100 1001 1001 1101 from binary to hexadecimal 499d

   b. (2 pts) Convert 1 from base 8 to binary 001

   c. (2 pts) Convert 001 000 101 from base 2 to base 8 105

   d. (2 pts) Convert 31 from base 8 to base 2 011 001

   e. (2 pts) Convert 3df7 from base 16 to binary 0011 1101 1111 0111

   f. (2 pts) Convert 23cf from base 16 to base 2 0010 0011 1100 1111

   g. (2 pts) Convert 0110 1001 0011 1111 from binary to hexadecimal 693f

   h. (2 pts) Convert 11 from base 8 to base 2 001 001

   i. (2 pts) Convert b6b1 from base 16 to base 2 1011 0110 1011 0001

   j. (2 pts) Convert 47 from base 8 to base 2 100 111

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  Node a;
  double b;
  int c;
  char d;
  Node *e;
  double *f;
  int *g;
  char *h;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) argv[1][2] *char*

b. (2 pts) *f *double*

c. (2 pts) e->next *Node **

d. (2 pts) e->next->next *Node **

e. (2 pts) argv[0] *char **

f. (2 pts) &h *char ***

g. (2 pts) c *int*

h. (2 pts) e *Node **

i. (2 pts) &b *double **

j. (2 pts) e->data *int*

k. (2 pts) argc *int*

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt guava grape
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of argv[1][0]? g

c. (2 pts) What is the value of argv[2][0]? g

d. (2 pts) What is the value of argv[0][4]? n

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 7aff from base 16 to base 2 0111 1010 1111 1111

   b. (2 pts) Convert 1101 0101 1101 1000 from base 2 to base 16 d5d8

   c. (2 pts) Convert 5925 from base 16 to base 2 0101 1001 0010 0101

   d. (2 pts) Convert 110 000 100 from base 2 to octal 604

   e. (2 pts) Convert 1001 0100 0001 0100 from base 2 to hexadecimal 9414

   f. (2 pts) Convert 0101 0110 0011 1000 from binary to hexadecimal 5638

   g. (2 pts) Convert 6b6d from base 16 to base 2 0110 1011 0110 1101

   h. (2 pts) Convert 43 from octal to base 2 100 011

   i. (2 pts) Convert 0101 1000 0110 1100 from binary to base 16 586c

   j. (2 pts) Convert 62 from base 8 to binary 110 010

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double a;
  int b;
  Node c;
  char d;
  double *e;
  int *f;
  Node *g;
  char *h;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) &e double **

b. (2 pts) d char

c. (2 pts) argv[0] char *

d. (2 pts) g->next Node *

e. (2 pts) &a double *

f. (2 pts) h char *

g. (2 pts) argc int

h. (2 pts) *f int

i. (2 pts) g->data int

j. (2 pts) g->next->next Node *

k. (2 pts) argv[1][2] char

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt fig apple date banana
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of argv[0][5]? I

c. (2 pts) What is the value of argv[1][2]? g

d. (2 pts) What is the value of argv[2][3]? l

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 57 from base 8 to base 2 101 111

   b. (2 pts) Convert 1001 0000 from binary to decimal 144

   c. (2 pts) Convert 0100 0111 from binary to base 10 71

   d. (2 pts) Convert 010 011 100 from base 2 to base 8 234

   e. (2 pts) Convert 011 101 110 from base 2 to base 8 356

   f. (2 pts) Convert 74 from base 8 to binary 111 100

   g. (2 pts) Convert 155d from hexadecimal to base 2 0001 0101 0101 1101

   h. (2 pts) Convert 22 from base 10 to base 2 0001 0110

   i. (2 pts) Convert 2283 from base 16 to base 2 0010 0010 1000 0011

   j. (2 pts) Convert 31 from base 8 to binary 011 001

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  Node x;
  int y;
  double z;
  char a;
  Node *b;
  int *c;
  double *d;
  char *e;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) *d double

b. (2 pts) argv[1][2] char

c. (2 pts) b->data int

d. (2 pts) &e char **

e. (2 pts) b->next Node *

f. (2 pts) &a char *

g. (2 pts) d double *

h. (2 pts) z double

i. (2 pts) argc int

j. (2 pts) argv[0] char *

k. (2 pts) b->next->next Node *

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt grape apple mango
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of argv[1][2]? a

c. (2 pts) What is the value of argv[0][0]? .

d. (2 pts) What is the value of argv[2][1]? p

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 5207 from hexadecimal to binary 0101 0010 0000 0111

   b. (2 pts) Convert 1111 0010 0100 0000 from binary to hexadecimal f240

   c. (2 pts) Convert 0100 1001 from base 2 to decimal 73

   d. (2 pts) Convert 45b8 from hexadecimal to binary 0100 0101 1011 1000

   e. (2 pts) Convert 3358 from hexadecimal to binary 0011 0011 0101 1000

   f. (2 pts) Convert c3b5 from hexadecimal to binary 1100 0011 1011 0101

   g. (2 pts) Convert 0111 0110 from binary to decimal 118

   h. (2 pts) Convert c3ce from base 16 to base 2 1100 0011 1100 1110

   i. (2 pts) Convert 9d33 from hexadecimal to base 2 1001 1101 0011 0011

   j. (2 pts) Convert 1010 1101 0110 1111 from binary to base 16 ad6f

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  Node e;
  double f;
  int g;
  char h;
  Node *p;
  double *q;
  int *r;
  char *s;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) argc int

b. (2 pts) &s char **

c. (2 pts) *r int

d. (2 pts) p->data int

e. (2 pts) s char *

f. (2 pts) p->next->next Node *

g. (2 pts) &g int *

h. (2 pts) argv[0] char *

i. (2 pts) argv[1][2] char

j. (2 pts) p->next Node *

k. (2 pts) g int

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt apple cherry
```

a. (2 pts) What is the value of `argc` in this case? *3*

b. (2 pts) What is the value of argv[1][1]? *p*

c. (2 pts) What is the value of argv[0][2]? *r*

d. (2 pts) What is the value of argv[2][5]? *y*

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 100 101 101 from base 2 to octal 455

   b. (2 pts) Convert 1010 1100 from base 2 to decimal 172

   c. (2 pts) Convert 55 from base 10 to binary 0011 0111

   d. (2 pts) Convert d1a2 from hexadecimal to binary 1101 0001 1010 0010

   e. (2 pts) Convert 5 from octal to base 2 101

   f. (2 pts) Convert 5d92 from base 16 to binary 0101 1101 1001 0010

   g. (2 pts) Convert 0010 0000 from base 2 to decimal 32

   h. (2 pts) Convert 22 from base 8 to base 2 010 010

   i. (2 pts) Convert 674a from hexadecimal to binary 0110 0111 0100 1010

   j. (2 pts) Convert 0100 1001 0101 1001 from base 2 to base 16 4959

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double a;
  int b;
  Node c;
  char d;
  double *e;
  int *f;
  Node *g;
  char *h;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) argv[0] *char **

b. (2 pts) &g *Node ***

c. (2 pts) g->next->next *Node **

d. (2 pts) argv[1][2] *char*

e. (2 pts) g->next *Node **

f. (2 pts) g *Node **

g. (2 pts) g->data *int*

h. (2 pts) &b *int **

i. (2 pts) argc *int*

j. (2 pts) a *double*

k. (2 pts) *e *double*

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt lime banana date apple
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of argv[0][5]? I

c. (2 pts) What is the value of argv[1][2]? m

d. (2 pts) What is the value of argv[2][3]? a

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 1100 1000 0100 1001 from binary to base 16 c849

   b. (2 pts) Convert 011 111 011 from base 2 to octal 373

   c. (2 pts) Convert 6dd5 from base 16 to base 2 0110 1101 1101 0101

   d. (2 pts) Convert 0010 1101 1110 0100 from base 2 to hexadecimal 2de4

   e. (2 pts) Convert 011 011 001 from binary to base 8 331

   f. (2 pts) Convert 43 from octal to base 2 100 011

   g. (2 pts) Convert 0010 0011 0001 1110 from binary to base 16 231e

   h. (2 pts) Convert 55 from octal to base 2 101 101

   i. (2 pts) Convert 905 from hexadecimal to binary 1001 0000 0101

   j. (2 pts) Convert 35 from base 8 to base 2 011 101

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int b;
  Node c;
  double d;
  char e;
  int *f;
  Node *g;
  double *h;
  char *p;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) &b *int **

b. (2 pts) *p *char*

c. (2 pts) c *Node*

d. (2 pts) g *Node **

e. (2 pts) g->data *int*

f. (2 pts) argv[0] *char **

g. (2 pts) &p *char ***

h. (2 pts) g->next->next *Node **

i. (2 pts) g->next *Node **

j. (2 pts) argv[1][2] *char*

k. (2 pts) argc *int*

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt kiwi guava cherry
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of argv[2][0]? g

c. (2 pts) What is the value of argv[0][3]? u

d. (2 pts) What is the value of argv[1][1]? i

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 1101 from binary to base 10 13

   b. (2 pts) Convert 3818 from hexadecimal to base 2 0011 1000 0001 1000

   c. (2 pts) Convert 5bb6 from hexadecimal to binary 0101 1011 1011 0110

   d. (2 pts) Convert 1011 1001 1100 1110 from binary to base 16 b9ce

   e. (2 pts) Convert 80 from decimal to base 2 0101 0000

   f. (2 pts) Convert 0010 1001 1101 1000 from binary to hexadecimal 29d8

   g. (2 pts) Convert 63 from base 8 to binary 110 011

   h. (2 pts) Convert 001 111 000 from binary to base 8 170

   i. (2 pts) Convert 1101 0011 0001 1100 from base 2 to hexadecimal d31c

   j. (2 pts) Convert 0001 0001 1000 0100 from binary to base 16 1184

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int y;
  Node z;
  double a;
  char b;
  int *c;
  Node *d;
  double *e;
  char *f;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) argv[0] *char **

b. (2 pts) &b *char **

c. (2 pts) *d *Node*

d. (2 pts) d->data *int*

e. (2 pts) &d *Node ***

f. (2 pts) argc *int*

g. (2 pts) y *int*

h. (2 pts) c *int **

i. (2 pts) d->next->next *Node **

j. (2 pts) argv[1][2] *char*

k. (2 pts) d->next *Node **

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt cherry date mango
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of argv[0][2]? r

c. (2 pts) What is the value of argv[2][1]? a

d. (2 pts) What is the value of argv[1][0]? c

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 0110 0101 from binary to base 10 *101*

   b. (2 pts) Convert 66 from octal to base 2 *110 110*

   c. (2 pts) Convert 0001 0001 1000 from base 2 to base 16 *118*

   d. (2 pts) Convert 117 from decimal to base 2 *0111 0101*

   e. (2 pts) Convert 1100 0000 1000 0101 from binary to hexadecimal *c085*

   f. (2 pts) Convert 001 010 110 from binary to octal *126*

   g. (2 pts) Convert 0001 1110 from binary to base 10 *30*

   h. (2 pts) Convert e28f from hexadecimal to binary *1110 0010 1000 1111*

   i. (2 pts) Convert 1100 0101 1000 1111 from base 2 to base 16 *c58f*

   j. (2 pts) Convert 29 from base 10 to binary *0001 1101*

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  Node p;
  double q;
  int r;
  char s;
  Node *t;
  double *w;
  int *x;
  char *y;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) argv[1][2] *char*

b. (2 pts) argv[0] *char **

c. (2 pts) t->next *Node **

d. (2 pts) argc *int*

e. (2 pts) *t *Node*

f. (2 pts) t->data *int*

g. (2 pts) t->next->next *Node **

h. (2 pts) s *char*

i. (2 pts) y *char **

j. (2 pts) &t *Node ***

k. (2 pts) &r *int **

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt cherry lemon
```

a. (2 pts) What is the value of `argc` in this case? *3*

b. (2 pts) What is the value of argv[1][3]? *r*

c. (2 pts) What is the value of argv[2][4]? *n*

d. (2 pts) What is the value of argv[0][0]? *.*

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 1010 1010 0110 0100 from binary to hexadecimal aa64

   b. (2 pts) Convert 149 from decimal to binary 1001 0101

   c. (2 pts) Convert 238 from decimal to base 2 1110 1110

   d. (2 pts) Convert 010 from binary to octal 2

   e. (2 pts) Convert 101 000 111 from binary to base 8 507

   f. (2 pts) Convert 1100 0101 from base 2 to decimal 197

   g. (2 pts) Convert 1100 1000 from base 2 to base 10 200

   h. (2 pts) Convert 32 from base 8 to binary 011 010

   i. (2 pts) Convert 8fa6 from hexadecimal to binary 1000 1111 1010 0110

   j. (2 pts) Convert 1011 1001 from binary to decimal 185

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int x;
  double y;
  Node z;
  char a;
  int *b;
  double *c;
  Node *d;
  char *e;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) d->next->next *Node **

b. (2 pts) argv[0] *char **

c. (2 pts) d *Node **

d. (2 pts) argv[1][2] *char*

e. (2 pts) d->data *int*

f. (2 pts) &d *Node ***

g. (2 pts) *d *Node*

h. (2 pts) argc *int*

i. (2 pts) y *double*

j. (2 pts) &z *Node **

k. (2 pts) d->next *Node **

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt cherry lime banana guava
```

a. (2 pts) What is the value of `argc` in this case? *5*

b. (2 pts) What is the value of argv[1][3]? *r*

c. (2 pts) What is the value of argv[0][2]? *r*

d. (2 pts) What is the value of argv[2][3]? *e*

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 219 from base 10 to binary 1101 1011

   b. (2 pts) Convert 0110 0110 from base 2 to base 10 102

   c. (2 pts) Convert 001 001 010 from binary to octal 112

   d. (2 pts) Convert 0101 1101 from binary to base 10 93

   e. (2 pts) Convert 1111 1001 from binary to base 10 249

   f. (2 pts) Convert 247 from base 10 to base 2 1111 0111

   g. (2 pts) Convert 1100 1010 from binary to decimal 202

   h. (2 pts) Convert 110 100 110 from binary to base 8 646

   i. (2 pts) Convert 49 from base 10 to base 2 0011 0001

   j. (2 pts) Convert 229 from decimal to base 2 1110 0101

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  Node x;
  int y;
  double z;
  char a;
  Node *b;
  int *c;
  double *d;
  char *e;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) b->data int

b. (2 pts) &b Node **

c. (2 pts) b->next Node *

d. (2 pts) argv[0] char *

e. (2 pts) argc int

f. (2 pts) c int *

g. (2 pts) &x Node *

h. (2 pts) b->next->next Node *

i. (2 pts) *c int

j. (2 pts) a char

k. (2 pts) argv[1][2] char

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt grape lime lemon
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of argv[2][3]? e

c. (2 pts) What is the value of argv[0][6]? t

d. (2 pts) What is the value of argv[1][1]? r

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 10 from base 8 to base 2 001 000

   b. (2 pts) Convert 0010 0000 1011 1011 from binary to hexadecimal 20bb

   c. (2 pts) Convert 0001 0011 0011 0111 from base 2 to hexadecimal 1337

   d. (2 pts) Convert 233 from decimal to binary 1110 1001

   e. (2 pts) Convert 221 from decimal to binary 1101 1101

   f. (2 pts) Convert 1001 0001 from binary to base 10 145

   g. (2 pts) Convert 0111 0100 from base 2 to base 10 116

   h. (2 pts) Convert 90 from decimal to binary 0101 1010

   i. (2 pts) Convert fb78 from hexadecimal to binary 1111 1011 0111 1000

   j. (2 pts) Convert 129 from base 10 to binary 1000 0001

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  Node s;
  int t;
  double w;
  char x;
  Node *y;
  int *z;
  double *a;
  char *b;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) t int

b. (2 pts) y->next Node *

c. (2 pts) y->data int

d. (2 pts) argc int

e. (2 pts) argv[1][2] char

f. (2 pts) &x char *

g. (2 pts) argv[0] char *

h. (2 pts) *b char

i. (2 pts) &a double **

j. (2 pts) y Node *

k. (2 pts) y->next->next Node *

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt date kiwi grape
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of argv[2][3]? i

c. (2 pts) What is the value of argv[1][2]? t

d. (2 pts) What is the value of argv[0][3]? u

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 36 from base 8 to base 2 011 110

   b. (2 pts) Convert c387 from hexadecimal to base 2 1100 0011 1000 0111

   c. (2 pts) Convert 56 from octal to base 2 101 110

   d. (2 pts) Convert 51 from octal to binary 101 001

   e. (2 pts) Convert 0100 1101 from binary to base 10 77

   f. (2 pts) Convert 44 from octal to base 2 100 100

   g. (2 pts) Convert 1100 0110 1000 0100 from binary to hexadecimal c684

   h. (2 pts) Convert 151 from hexadecimal to binary 0001 0101 0001

   i. (2 pts) Convert edec from base 16 to base 2 1110 1101 1110 1100

   j. (2 pts) Convert 1000 1100 from binary to base 10 140

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double e;
  Node f;
  int g;
  char h;
  double *p;
  Node *q;
  int *r;
  char *s;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) argv[1][2] *char*

b. (2 pts) q->next->next *Node **

c. (2 pts) &h *char **

d. (2 pts) *r *int*

e. (2 pts) e *double*

f. (2 pts) q->next *Node **

g. (2 pts) s *char **

h. (2 pts) &q *Node ***

i. (2 pts) q->data *int*

j. (2 pts) argc *int*

k. (2 pts) argv[0] *char **

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt banana lemon
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of argv[2][4]? n

c. (2 pts) What is the value of argv[1][5]? a

d. (2 pts) What is the value of argv[0][1]? /

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 1011 1101 from base 2 to base 10 189

   b. (2 pts) Convert 0111 1101 1010 1111 from binary to hexadecimal 7daf

   c. (2 pts) Convert 51 from base 8 to binary 101 001

   d. (2 pts) Convert 0011 0000 1111 0100 from binary to base 16 30f4

   e. (2 pts) Convert 30fe from hexadecimal to binary 0011 0000 1111 1110

   f. (2 pts) Convert 0010 1100 0111 1001 from binary to hexadecimal 2c79

   g. (2 pts) Convert 0111 0000 0111 0101 from base 2 to hexadecimal 7075

   h. (2 pts) Convert 100 010 010 from binary to octal 422

   i. (2 pts) Convert 184 from decimal to binary 1011 1000

   j. (2 pts) Convert 0010 1000 from base 2 to base 10 40

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double b;
  int c;
  Node d;
  char e;
  double *f;
  int *g;
  Node *h;
  char *p;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) argv[0] *char **

b. (2 pts) g *int **

c. (2 pts) &d *Node **

d. (2 pts) *f *double*

e. (2 pts) h->next *Node **

f. (2 pts) e *char*

g. (2 pts) &p *char ***

h. (2 pts) h->next->next *Node **

i. (2 pts) argc *int*

j. (2 pts) argv[1][2] *char*

k. (2 pts) h->data *int*

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt grape guava banana fig
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of argv[2][1]? u

c. (2 pts) What is the value of argv[0][5]? I

d. (2 pts) What is the value of argv[1][1]? r

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 111 011 110 from binary to octal 736

   b. (2 pts) Convert 23 from base 8 to base 2 010 011

   c. (2 pts) Convert 1101 1100 from base 2 to base 10 220

   d. (2 pts) Convert 1000 1101 from binary to base 10 141

   e. (2 pts) Convert 1000 0111 from binary to decimal 135

   f. (2 pts) Convert 010 111 101 from base 2 to base 8 275

   g. (2 pts) Convert 72a3 from hexadecimal to binary 0111 0010 1010 0011

   h. (2 pts) Convert 111 100 011 from base 2 to octal 743

   i. (2 pts) Convert 0101 1001 1011 1101 from base 2 to base 16 59bd

   j. (2 pts) Convert 0101 0100 from base 2 to decimal 84

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int s;
  Node t;
  double w;
  char x;
  int *y;
  Node *z;
  double *a;
  char *b;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) z->next->next *Node **

b. (2 pts) argv[0] *char **

c. (2 pts) argc *int*

d. (2 pts) &y *int ***

e. (2 pts) z->next *Node **

f. (2 pts) z->data *int*

g. (2 pts) z *Node **

h. (2 pts) &t *Node **

i. (2 pts) argv[1][2] *char*

j. (2 pts) *b *char*

k. (2 pts) s *int*

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt lime kiwi apple
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of argv[0][0]? .

c. (2 pts) What is the value of argv[2][1]? i

d. (2 pts) What is the value of argv[1][3]? e

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 52 from base 10 to binary 0011 0100

   b. (2 pts) Convert 010 010 from base 2 to octal 22

   c. (2 pts) Convert 1100 1010 from base 2 to base 10 202

   d. (2 pts) Convert 6 from octal to binary 110

   e. (2 pts) Convert 6a68 from hexadecimal to binary 0110 1010 0110 1000

   f. (2 pts) Convert 111 110 001 from binary to base 8 761

   g. (2 pts) Convert 0001 1100 1001 0100 from base 2 to hexadecimal 1c94

   h. (2 pts) Convert 121 from base 10 to binary 0111 1001

   i. (2 pts) Convert 35 from decimal to binary 0010 0011

   j. (2 pts) Convert 1111 0000 from base 2 to decimal 240

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  Node h;
  int p;
  double q;
  char r;
  Node *s;
  int *t;
  double *w;
  char *x;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) argc *int*

b. (2 pts) s *Node **

c. (2 pts) &h *Node **

d. (2 pts) s->next->next *Node **

e. (2 pts) q *double*

f. (2 pts) argv[1][2] *char*

g. (2 pts) *t *int*

h. (2 pts) s->next *Node **

i. (2 pts) s->data *int*

j. (2 pts) argv[0] *char **

k. (2 pts) &x *char ***

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt lime fig cherry
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of argv[1][2]? m

c. (2 pts) What is the value of argv[2][0]? f

d. (2 pts) What is the value of argv[0][6]? t

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 8c7f from hexadecimal to binary 1000 1100 0111 1111

   b. (2 pts) Convert 1010 1100 0010 1010 from binary to base 16 ac2a

   c. (2 pts) Convert 112 from base 10 to binary 0111 0000

   d. (2 pts) Convert d4b3 from base 16 to binary 1101 0100 1011 0011

   e. (2 pts) Convert 218 from decimal to base 2 1101 1010

   f. (2 pts) Convert 1111 1010 from binary to base 10 250

   g. (2 pts) Convert 0110 1110 0100 0111 from binary to base 16 6e47

   h. (2 pts) Convert 10 from base 8 to binary 001 000

   i. (2 pts) Convert 22 from base 10 to base 2 0001 0110

   j. (2 pts) Convert 111 111 000 from base 2 to base 8 770

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  Node a;
  double b;
  int c;
  char d;
  Node *e;
  double *f;
  int *g;
  char *h;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) argv[0] *char **

b. (2 pts) &f *double ***

c. (2 pts) argv[1][2] *char*

d. (2 pts) *e *Node*

e. (2 pts) e->next->next *Node **

f. (2 pts) &d *char **

g. (2 pts) e->data *int*

h. (2 pts) argc *int*

i. (2 pts) g *int **

j. (2 pts) b *double*

k. (2 pts) e->next *Node **

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt apple fig
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of argv[0][0]? .

c. (2 pts) What is the value of argv[1][1]? p

d. (2 pts) What is the value of argv[2][0]? f

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 64 from base 8 to base 2 110 100

   b. (2 pts) Convert 0110 0110 from binary to base 10 102

   c. (2 pts) Convert 010 111 011 from base 2 to octal 273

   d. (2 pts) Convert 609d from base 16 to base 2 0110 0000 1001 1101

   e. (2 pts) Convert be2c from hexadecimal to binary 1011 1110 0010 1100

   f. (2 pts) Convert 147 from decimal to base 2 1001 0011

   g. (2 pts) Convert 110 000 from base 2 to base 8 60

   h. (2 pts) Convert 101 001 111 from binary to octal 517

   i. (2 pts) Convert e05f from hexadecimal to binary 1110 0000 0101 1111

   j. (2 pts) Convert 100 110 000 from binary to octal 460

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double x;
  int y;
  Node z;
  char a;
  double *b;
  int *c;
  Node *d;
  char *e;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) argc *int*

b. (2 pts) argv[0] *char **

c. (2 pts) d->next *Node **

d. (2 pts) *d *Node*

e. (2 pts) d->data *int*

f. (2 pts) argv[1][2] *char*

g. (2 pts) &b *double***

h. (2 pts) &a *char **

i. (2 pts) d->next->next *Node **

j. (2 pts) c *int **

k. (2 pts) x *double*

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt cherry grape apple mango
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of argv[1][2]? e

c. (2 pts) What is the value of argv[0][3]? u

d. (2 pts) What is the value of argv[2][0]? g

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 0010 1100 0000 from binary to hexadecimal 2c0

   b. (2 pts) Convert 001 101 111 from base 2 to base 8 157

   c. (2 pts) Convert 945a from base 16 to binary 1001 0100 0101 1010

   d. (2 pts) Convert 1011 1100 1101 1110 from base 2 to hexadecimal bcde

   e. (2 pts) Convert 0001 0100 0100 1010 from binary to hexadecimal 144a

   f. (2 pts) Convert c656 from base 16 to binary 1100 0110 0101 0110

   g. (2 pts) Convert 0001 1010 0110 0110 from binary to base 16 1a66

   h. (2 pts) Convert 16 from base 10 to binary 0001 0000

   i. (2 pts) Convert 821a from base 16 to base 2 1000 0010 0001 1010

   j. (2 pts) Convert 110 001 001 from binary to base 8 611

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int h;
  double p;
  Node q;
  char r;
  int *s;
  double *t;
  Node *w;
  char *x;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) w->next->next *Node **

b. (2 pts) argv[0] *char **

c. (2 pts) argv[1][2] *char*

d. (2 pts) argc *int*

e. (2 pts) &t *double ***

f. (2 pts) p *double*

g. (2 pts) s *int **

h. (2 pts) w->next *Node **

i. (2 pts) *t *double*

j. (2 pts) w->data *int*

k. (2 pts) &p *double **

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt date guava apple fig
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of argv[0][4]? n

c. (2 pts) What is the value of argv[2][3]? v

d. (2 pts) What is the value of argv[1][0]? d

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 010 001 111 from binary to octal 217

   b. (2 pts) Convert 1111 0010 from base 2 to decimal 242

   c. (2 pts) Convert 130 from decimal to binary 1000 0010

   d. (2 pts) Convert 0100 1000 1100 1000 from base 2 to base 16 48c8

   e. (2 pts) Convert 111 101 111 from binary to base 8 757

   f. (2 pts) Convert 96 from base 10 to base 2 0110 0000

   g. (2 pts) Convert 1100 0100 0101 0110 from base 2 to base 16 c456

   h. (2 pts) Convert 9862 from base 16 to binary 1001 1000 0110 0010

   i. (2 pts) Convert 76 from decimal to binary 0100 1100

   j. (2 pts) Convert 011 000 001 from binary to base 8 301

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int e;
  Node f;
  double g;
  char h;
  int *p;
  Node *q;
  double *r;
  char *s;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) h *char*

b. (2 pts) *s *char*

c. (2 pts) q->data *int*

d. (2 pts) q->next->next *Node **

e. (2 pts) argv[0] *char **

f. (2 pts) argv[1][2] *char*

g. (2 pts) &s *char ***

h. (2 pts) s *char **

i. (2 pts) &e *int **

j. (2 pts) q->next *Node **

k. (2 pts) argc *int*

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt mango fig grape
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of argv[2][1]? i

c. (2 pts) What is the value of argv[0][4]? n

d. (2 pts) What is the value of argv[1][3]? g

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 001 000 from base 2 to base 8 *10*

   b. (2 pts) Convert 79 from base 10 to base 2 *0100 1111*

   c. (2 pts) Convert 011 010 111 from binary to base 8 *327*

   d. (2 pts) Convert 41 from octal to binary *100 001*

   e. (2 pts) Convert 77 from octal to binary *111 111*

   f. (2 pts) Convert 0010 0101 from base 2 to base 10 *37*

   g. (2 pts) Convert 65 from base 8 to binary *110 101*

   h. (2 pts) Convert 010 010 001 from base 2 to base 8 *221*

   i. (2 pts) Convert 1111 1011 1100 0001 from base 2 to hexadecimal *fbc1*

   j. (2 pts) Convert 1110 1110 from base 2 to decimal *238*

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double c;
  Node d;
  int e;
  char f;
  double *g;
  Node *h;
  int *p;
  char *q;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) h->next Node *

b. (2 pts) h->next->next Node *

c. (2 pts) h->data int

d. (2 pts) h Node *

e. (2 pts) argc int

f. (2 pts) argv[0] char *

g. (2 pts) d Node

h. (2 pts) *q char

i. (2 pts) argv[1][2] char

j. (2 pts) &h Node **

k. (2 pts) &c double *

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt date lime
```

a. (2 pts) What is the value of `argc` in this case? *3*

b. (2 pts) What is the value of argv[2][2]? *m*

c. (2 pts) What is the value of argv[1][3]? *e*

d. (2 pts) What is the value of argv[0][6]? *t*

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 48f2 from base 16 to binary 0100 1000 1111 0010

   b. (2 pts) Convert 2 from octal to binary 010

   c. (2 pts) Convert 010 110 010 from base 2 to base 8 262

   d. (2 pts) Convert 100 101 from binary to base 8 45

   e. (2 pts) Convert 111 000 011 from binary to base 8 703

   f. (2 pts) Convert 101 111 111 from binary to octal 577

   g. (2 pts) Convert 40 from octal to binary 100 000

   h. (2 pts) Convert 110 100 000 from base 2 to base 8 640

   i. (2 pts) Convert 1100 0101 1101 1000 from base 2 to base 16 c5d8

   j. (2 pts) Convert 1000 1010 from base 2 to decimal 138

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double h;
  int p;
  Node q;
  char r;
  double *s;
  int *t;
  Node *w;
  char *x;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) w->data *int*

b. (2 pts) &x *char ***

c. (2 pts) s *double **

d. (2 pts) w->next *Node **

e. (2 pts) w->next->next *Node **

f. (2 pts) argv[1][2] *char*

g. (2 pts) &r *char **

h. (2 pts) h *double*

i. (2 pts) argv[0] *char **

j. (2 pts) *s *double*

k. (2 pts) argc *int*

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt grape guava kiwi lime
```

a. (2 pts) What is the value of `argc` in this case? *5*

b. (2 pts) What is the value of argv[1][3]? *p*

c. (2 pts) What is the value of argv[0][0]? *.*

d. (2 pts) What is the value of argv[2][3]? *v*

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 0111 1010 from binary to decimal 122

   b. (2 pts) Convert 1101 1011 0110 0010 from binary to base 16 db62

   c. (2 pts) Convert 1000 1111 1100 1111 from base 2 to hexadecimal 8fcf

   d. (2 pts) Convert 0110 1111 from binary to decimal 111

   e. (2 pts) Convert 0011 1000 from binary to base 10 56

   f. (2 pts) Convert f22d from base 16 to base 2 1111 0010 0010 1101

   g. (2 pts) Convert 1000 0100 0001 0100 from base 2 to base 16 8414

   h. (2 pts) Convert 3967 from hexadecimal to base 2 0011 1001 0110 0111

   i. (2 pts) Convert 0110 0111 1001 0011 from base 2 to hexadecimal 6793

   j. (2 pts) Convert 182 from base 10 to binary 1011 0110

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int p;
  Node q;
  double r;
  char s;
  int *t;
  Node *w;
  double *x;
  char *y;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) w->next->next Node *

b. (2 pts) argc int

c. (2 pts) &t int **

d. (2 pts) w->next Node *

e. (2 pts) argv[1][2] char

f. (2 pts) *t int

g. (2 pts) q Node

h. (2 pts) &r double *

i. (2 pts) w->data int

j. (2 pts) y char *

k. (2 pts) argv[0] char *

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt lemon grape date
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of argv[2][3]? p

c. (2 pts) What is the value of argv[0][6]? t

d. (2 pts) What is the value of argv[1][3]? o

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 1011 1111 0011 0011 from binary to base 16 bf33

   b. (2 pts) Convert 45 from base 8 to binary 100 101

   c. (2 pts) Convert 125 from decimal to binary 0111 1101

   d. (2 pts) Convert 76 from octal to base 2 111 110

   e. (2 pts) Convert 1b57 from hexadecimal to base 2 0001 1011 0101 0111

   f. (2 pts) Convert 140 from base 10 to base 2 1000 1100

   g. (2 pts) Convert 0010 1110 0000 0101 from binary to base 16 2e05

   h. (2 pts) Convert c118 from hexadecimal to base 2 1100 0001 0001 1000

   i. (2 pts) Convert 14 from octal to base 2 001 100

   j. (2 pts) Convert 0101 0010 from binary to decimal 82

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  Node f;
  double g;
  int h;
  char p;
  Node *q;
  double *r;
  int *s;
  char *t;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) q->next Node *

b. (2 pts) argv[0] char *

c. (2 pts) &t char **

d. (2 pts) q->data int

e. (2 pts) &g double *

f. (2 pts) argc int

g. (2 pts) *r double

h. (2 pts) argv[1][2] char

i. (2 pts) p char

j. (2 pts) s int *

k. (2 pts) q->next->next Node *

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt guava grape
```

a. (2 pts) What is the value of `argc` in this case? *3*

b. (2 pts) What is the value of argv[2][0]? *g*

c. (2 pts) What is the value of argv[1][2]? *a*

d. (2 pts) What is the value of argv[0][2]? *r*

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 0001 0111 1001 0000 from binary to base 16 *1790*

   b. (2 pts) Convert 3857 from base 16 to base 2 *0011 1000 0101 0111*

   c. (2 pts) Convert 2312 from base 16 to base 2 *0010 0011 0001 0010*

   d. (2 pts) Convert b6be from base 16 to base 2 *1011 0110 1011 1110*

   e. (2 pts) Convert 42 from octal to base 2 *100 010*

   f. (2 pts) Convert 1000 1101 0101 1010 from base 2 to base 16 *8d5a*

   g. (2 pts) Convert 0111 1111 from base 2 to base 10 *127*

   h. (2 pts) Convert 011 001 111 from binary to base 8 *317*

   i. (2 pts) Convert 11 from octal to binary *001 001*

   j. (2 pts) Convert 010 111 100 from binary to base 8 *274*

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  Node y;
  double z;
  int a;
  char b;
  Node *c;
  double *d;
  int *e;
  char *f;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) b *char*

b. (2 pts) c *Node **

c. (2 pts) c->next->next *Node **

d. (2 pts) argc *int*

e. (2 pts) &d *double***

f. (2 pts) c->next *Node **

g. (2 pts) c->data *int*

h. (2 pts) argv[0] *char **

i. (2 pts) &y *Node **

j. (2 pts) argv[1][2] *char*

k. (2 pts) *d *double*

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt cherry lemon
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of argv[0][5]? I

c. (2 pts) What is the value of argv[1][1]? h

d. (2 pts) What is the value of argv[2][3]? o

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 010 111 000 from binary to base 8 270

   b. (2 pts) Convert 111 100 101 from base 2 to base 8 745

   c. (2 pts) Convert 0001 0000 from binary to decimal 16

   d. (2 pts) Convert 42a8 from base 16 to binary 0100 0010 1010 1000

   e. (2 pts) Convert 0110 1111 from binary to base 10 111

   f. (2 pts) Convert 11 from octal to base 2 001 001

   g. (2 pts) Convert 0010 1001 from binary to base 10 41

   h. (2 pts) Convert 239 from base 10 to base 2 1110 1111

   i. (2 pts) Convert 1110 1110 0011 0101 from base 2 to base 16 ee35

   j. (2 pts) Convert 111 110 100 from binary to base 8 764

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int t;
  double w;
  Node x;
  char y;
  int *z;
  double *a;
  Node *b;
  char *c;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) w *double*

b. (2 pts) *b *Node*

c. (2 pts) b->next *Node **

d. (2 pts) argc *int*

e. (2 pts) argv[0] *char **

f. (2 pts) b->data *int*

g. (2 pts) c *char **

h. (2 pts) &z *int ***

i. (2 pts) &t *int **

j. (2 pts) b->next->next *Node **

k. (2 pts) argv[1][2] *char*

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt date guava lemon mango
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of argv[1][1]? a

c. (2 pts) What is the value of argv[0][5]? I

d. (2 pts) What is the value of argv[2][1]? u

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 43 from base 8 to binary 100 011

   b. (2 pts) Convert 61 from octal to base 2 110 001

   c. (2 pts) Convert 21 from octal to base 2 010 001

   d. (2 pts) Convert 1001 1110 1110 1001 from binary to hexadecimal 9ee9

   e. (2 pts) Convert 197 from decimal to base 2 1100 0101

   f. (2 pts) Convert 26 from octal to binary 010 110

   g. (2 pts) Convert 0010 1011 from base 2 to decimal 43

   h. (2 pts) Convert 5829 from base 16 to base 2 0101 1000 0010 1001

   i. (2 pts) Convert 43 from octal to binary 100 011

   j. (2 pts) Convert 0010 0110 from binary to decimal 38

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  Node e;
  int f;
  double g;
  char h;
  Node *p;
  int *q;
  double *r;
  char *s;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) g *double*

b. (2 pts) p->data *int*

c. (2 pts) argv[1][2] *char*

d. (2 pts) &g *double **

e. (2 pts) &q *int ***

f. (2 pts) s *char **

g. (2 pts) argv[0] *char **

h. (2 pts) p->next->next *Node **

i. (2 pts) argc *int*

j. (2 pts) *r *double*

k. (2 pts) p->next *Node **

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt grape lime fig
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of argv[0][3]? u

c. (2 pts) What is the value of argv[2][3]? e

d. (2 pts) What is the value of argv[1][1]? r

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 1101 0010 from binary to decimal 210

   b. (2 pts) Convert 126 from base 10 to binary 0111 1110

   c. (2 pts) Convert 3531 from hexadecimal to base 2 0011 0101 0011 0001

   d. (2 pts) Convert 0010 1010 1101 0100 from binary to hexadecimal 2ad4

   e. (2 pts) Convert 1010 1000 1000 0100 from binary to hexadecimal a884

   f. (2 pts) Convert 74 from base 8 to base 2 111 100

   g. (2 pts) Convert 1101 0101 from binary to decimal 213

   h. (2 pts) Convert 67 from octal to base 2 110 111

   i. (2 pts) Convert 0101 1010 0000 0110 from base 2 to base 16 5a06

   j. (2 pts) Convert 1100 0010 from base 2 to base 10 194

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  Node b;
  double c;
  int d;
  char e;
  Node *f;
  double *g;
  int *h;
  char *p;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) b *Node*

b. (2 pts) f->next->next *Node **

c. (2 pts) *p *char*

d. (2 pts) f->data *int*

e. (2 pts) f->next *Node **

f. (2 pts) argv[0] *char **

g. (2 pts) argv[1][2] *char*

h. (2 pts) &c *double **

i. (2 pts) &p *char ***

j. (2 pts) argc *int*

k. (2 pts) g *double **

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt banana fig
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of argv[2][2]? g

c. (2 pts) What is the value of argv[0][6]? t

d. (2 pts) What is the value of argv[1][4]? n

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 43 from base 10 to binary 0010 1011

   b. (2 pts) Convert 0010 0000 from base 2 to decimal 32

   c. (2 pts) Convert 1101 1010 1001 0011 from binary to hexadecimal da93

   d. (2 pts) Convert 230 from base 10 to base 2 1110 0110

   e. (2 pts) Convert 110 001 from binary to base 8 61

   f. (2 pts) Convert 244 from base 10 to base 2 1111 0100

   g. (2 pts) Convert 39 from base 10 to binary 0010 0111

   h. (2 pts) Convert 100 001 100 from binary to base 8 414

   i. (2 pts) Convert 0100 1100 0111 1010 from binary to hexadecimal 4c7a

   j. (2 pts) Convert 110 011 100 from base 2 to octal 634

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double s;
  Node t;
  int w;
  char x;
  double *y;
  Node *z;
  int *a;
  char *b;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) &t *Node **

b. (2 pts) z->next->next *Node **

c. (2 pts) s *double*

d. (2 pts) &a *int ***

e. (2 pts) *a *int*

f. (2 pts) argv[0] *char **

g. (2 pts) argc *int*

h. (2 pts) y *double **

i. (2 pts) argv[1][2] *char*

j. (2 pts) z->data *int*

k. (2 pts) z->next *Node **

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt lime mango
```

a. (2 pts) What is the value of `argc` in this case? *3*

b. (2 pts) What is the value of argv[2][0]? *m*

c. (2 pts) What is the value of argv[1][1]? *i*

d. (2 pts) What is the value of argv[0][6]? *t*

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 0110 1111 1110 1100 from binary to base 16 6fec

   b. (2 pts) Convert 219 from base 10 to base 2 1101 1011

   c. (2 pts) Convert 1100 1000 0111 0011 from binary to hexadecimal c873

   d. (2 pts) Convert 011 100 100 from base 2 to octal 344

   e. (2 pts) Convert 1111 1100 0100 1000 from binary to hexadecimal fc48

   f. (2 pts) Convert 1000 1110 from base 2 to base 10 142

   g. (2 pts) Convert 1101 0001 0110 1011 from base 2 to hexadecimal d16b

   h. (2 pts) Convert dfb from hexadecimal to base 2 1101 1111 1011

   i. (2 pts) Convert 1691 from base 16 to base 2 0001 0110 1001 0001

   j. (2 pts) Convert 011 010 100 from base 2 to octal 324

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int p;
  double q;
  Node r;
  char s;
  int *t;
  double *w;
  Node *x;
  char *y;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) &t *int* **

b. (2 pts) x->next *Node* *

c. (2 pts) *t *int*

d. (2 pts) x->next->next *Node* *

e. (2 pts) &p *int* *

f. (2 pts) argv[1][2] *char*

g. (2 pts) x->data *int*

h. (2 pts) argc *int*

i. (2 pts) argv[0] *char* *

j. (2 pts) y *char* *

k. (2 pts) r *Node*

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt lemon cherry kiwi mango
```

a. (2 pts) What is the value of `argc` in this case? *5*

b. (2 pts) What is the value of argv[1][1]? *e*

c. (2 pts) What is the value of argv[0][3]? *u*

d. (2 pts) What is the value of argv[2][4]? *r*

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert a14e from base 16 to binary 1010 0001 0100 1110

   b. (2 pts) Convert 1010 1100 from binary to decimal 172

   c. (2 pts) Convert 77 from octal to binary 111 111

   d. (2 pts) Convert ce92 from base 16 to base 2 1100 1110 1001 0010

   e. (2 pts) Convert 5266 from hexadecimal to binary 0101 0010 0110 0110

   f. (2 pts) Convert 1100 0001 0001 0100 from binary to hexadecimal c114

   g. (2 pts) Convert 1101 0011 from base 2 to base 10 211

   h. (2 pts) Convert 76ea from base 16 to base 2 0111 0110 1110 1010

   i. (2 pts) Convert 1011 1000 0100 1100 from binary to hexadecimal b84c

   j. (2 pts) Convert 100 101 100 from base 2 to base 8 454

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  Node a;
  int b;
  double c;
  char d;
  Node *e;
  int *f;
  double *g;
  char *h;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) argv[1][2] *char*

b. (2 pts) argc *int*

c. (2 pts) &d *char **

d. (2 pts) &g *double ***

e. (2 pts) e->data *int*

f. (2 pts) *h *char*

g. (2 pts) argv[0] *char **

h. (2 pts) d *char*

i. (2 pts) e->next->next *Node **

j. (2 pts) e->next *Node **

k. (2 pts) g *double **

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt mango guava fig
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of argv[0][1]? /

c. (2 pts) What is the value of argv[1][1]? a

d. (2 pts) What is the value of argv[2][4]? a

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 71 from base 8 to binary 111 001

   b. (2 pts) Convert 0110 0110 1101 0001 from binary to base 16 66d1

   c. (2 pts) Convert 111 011 001 from base 2 to octal 731

   d. (2 pts) Convert 90 from base 10 to binary 0101 1010

   e. (2 pts) Convert 0011 0101 1011 0010 from binary to hexadecimal 35b2

   f. (2 pts) Convert 5af1 from base 16 to binary 0101 1010 1111 0001

   g. (2 pts) Convert 0111 1101 from binary to base 10 125

   h. (2 pts) Convert 111 111 101 from base 2 to base 8 775

   i. (2 pts) Convert 8263 from base 16 to base 2 1000 0010 0110 0011

   j. (2 pts) Convert 001 100 100 from binary to base 8 144

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double x;
  Node y;
  int z;
  char a;
  double *b;
  Node *c;
  int *d;
  char *e;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) &z *int **

b. (2 pts) argv[0] *char **

c. (2 pts) c->next *Node **

d. (2 pts) &b *double ***

e. (2 pts) argc *int*

f. (2 pts) argv[1][2] *char*

g. (2 pts) c *Node **

h. (2 pts) c->next->next *Node **

i. (2 pts) c->data *int*

j. (2 pts) y *Node*

k. (2 pts) *c *Node*

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt apple lime
```

a. (2 pts) What is the value of `argc` in this case? *3*

b. (2 pts) What is the value of argv[1][3]? *l*

c. (2 pts) What is the value of argv[2][3]? *e*

d. (2 pts) What is the value of argv[0][2]? *r*

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 001 111 101 from base 2 to octal 175

   b. (2 pts) Convert 9 from base 10 to binary 1001

   c. (2 pts) Convert 1001 0010 from binary to base 10 146

   d. (2 pts) Convert 5 from base 8 to binary 101

   e. (2 pts) Convert a62b from hexadecimal to binary 1010 0110 0010 1011

   f. (2 pts) Convert 27 from base 8 to binary 010 111

   g. (2 pts) Convert 1100 1111 0011 1101 from binary to base 16 cf3d

   h. (2 pts) Convert 165 from base 10 to base 2 1010 0101

   i. (2 pts) Convert 74d6 from hexadecimal to binary 0111 0100 1101 0110

   j. (2 pts) Convert 17 from base 8 to binary 001 111

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double h;
  Node p;
  int q;
  char r;
  double *s;
  Node *t;
  int *w;
  char *x;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) &p Node *

b. (2 pts) argv[0] char *

c. (2 pts) t->data int

d. (2 pts) argc int

e. (2 pts) p Node

f. (2 pts) t->next Node *

g. (2 pts) argv[1][2] char

h. (2 pts) &x char **

i. (2 pts) s double *

j. (2 pts) *s double

k. (2 pts) t->next->next Node *

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt lime kiwi
```

a. (2 pts) What is the value of `argc` in this case? *3*

b. (2 pts) What is the value of argv[1][0]? *l*

c. (2 pts) What is the value of argv[2][2]? *w*

d. (2 pts) What is the value of argv[0][0]? *.*

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 131 from base 10 to binary 1000 0011

   b. (2 pts) Convert c3c6 from base 16 to base 2 1100 0011 1100 0110

   c. (2 pts) Convert 37 from base 8 to base 2 011 111

   d. (2 pts) Convert 101 000 011 from base 2 to base 8 503

   e. (2 pts) Convert 1000 1001 0111 0110 from binary to hexadecimal 8976

   f. (2 pts) Convert 111 101 100 from base 2 to base 8 754

   g. (2 pts) Convert 0111 1001 0010 1110 from base 2 to base 16 792e

   h. (2 pts) Convert 2cbd from hexadecimal to base 2 0010 1100 1011 1101

   i. (2 pts) Convert 0011 1110 1110 1101 from base 2 to base 16 3eed

   j. (2 pts) Convert 66 from octal to base 2 110 110

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double e;
  int f;
  Node g;
  char h;
  double *p;
  int *q;
  Node *r;
  char *s;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) argc *int*

b. (2 pts) &q *int* **

c. (2 pts) *r *Node*

d. (2 pts) s *char* *

e. (2 pts) r->data *int*

f. (2 pts) r->next->next *Node* *

g. (2 pts) argv[1][2] *char*

h. (2 pts) r->next *Node* *

i. (2 pts) h *char*

j. (2 pts) argv[0] *char* *

k. (2 pts) &f *int* *

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt lime apple lemon banana
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of argv[0][5]? I

c. (2 pts) What is the value of argv[1][0]? l

d. (2 pts) What is the value of argv[2][0]? a

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 1011 0100 from binary to base 10 180

   b. (2 pts) Convert 1001 0101 0100 1100 from binary to hexadecimal 954c

   c. (2 pts) Convert 182 from base 10 to base 2 1011 0110

   d. (2 pts) Convert 1111 1110 from base 2 to decimal 254

   e. (2 pts) Convert 67 from octal to binary 110 111

   f. (2 pts) Convert 0010 1000 from binary to decimal 40

   g. (2 pts) Convert 0111 1011 0101 1100 from binary to hexadecimal 7b5c

   h. (2 pts) Convert 45 from octal to base 2 100 101

   i. (2 pts) Convert 1110 0000 1010 1000 from binary to hexadecimal e0a8

   j. (2 pts) Convert 1 from octal to binary 001

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int f;
  Node g;
  double h;
  char p;
  int *q;
  Node *r;
  double *s;
  char *t;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) r->data *int*

b. (2 pts) s *double **

c. (2 pts) r->next *Node **

d. (2 pts) argv[0] *char **

e. (2 pts) argv[1][2] *char*

f. (2 pts) f *int*

g. (2 pts) argc *int*

h. (2 pts) &p *char **

i. (2 pts) *r *Node*

j. (2 pts) r->next->next *Node **

k. (2 pts) &s *double ***

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt guava mango apple
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of argv[0][5]? I

c. (2 pts) What is the value of argv[2][2]? n

d. (2 pts) What is the value of argv[1][0]? g

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert f9ab from base 16 to binary 1111 1001 1010 1011

   b. (2 pts) Convert 23 from octal to base 2 010 011

   c. (2 pts) Convert 101 001 000 from base 2 to base 8 510

   d. (2 pts) Convert 42 from base 8 to base 2 100 010

   e. (2 pts) Convert 1100 0010 from binary to base 10 194

   f. (2 pts) Convert 110 000 100 from binary to octal 604

   g. (2 pts) Convert 0010 0101 0100 1100 from base 2 to base 16 254c

   h. (2 pts) Convert 111 010 from base 2 to base 8 72

   i. (2 pts) Convert 52 from base 8 to base 2 101 010

   j. (2 pts) Convert 50 from base 8 to binary 101 000

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double s;
  Node t;
  int w;
  char x;
  double *y;
  Node *z;
  int *a;
  char *b;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) z->data int

b. (2 pts) argc int

c. (2 pts) z->next Node *

d. (2 pts) z->next->next Node *

e. (2 pts) &w int *

f. (2 pts) z Node *

g. (2 pts) x char

h. (2 pts) argv[1][2] char

i. (2 pts) &z Node **

j. (2 pts) *b char

k. (2 pts) argv[0] char *

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt fig apple
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of argv[2][0]? a

c. (2 pts) What is the value of argv[1][0]? f

d. (2 pts) What is the value of argv[0][3]? u

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 1110 1101 1111 0001 from base 2 to hexadecimal edf1

   b. (2 pts) Convert 55 from decimal to base 2 0011 0111

   c. (2 pts) Convert 0101 1010 0010 from binary to hexadecimal 5a2

   d. (2 pts) Convert 60 from base 8 to base 2 110 000

   e. (2 pts) Convert 1001 1100 from binary to decimal 156

   f. (2 pts) Convert 255 from decimal to base 2 1111 1111

   g. (2 pts) Convert 1011 0101 from base 2 to base 10 181

   h. (2 pts) Convert 185 from base 10 to base 2 1011 1001

   i. (2 pts) Convert 1110 0000 from binary to decimal 224

   j. (2 pts) Convert 001 010 101 from binary to base 8 125

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double h;
  int p;
  Node q;
  char r;
  double *s;
  int *t;
  Node *w;
  char *x;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) &t *int* **

b. (2 pts) s *double* *

c. (2 pts) w->data *int*

d. (2 pts) argv[0] *char* *

e. (2 pts) r *char*

f. (2 pts) argc *int*

g. (2 pts) argv[1][2] *char*

h. (2 pts) &q *Node* *

i. (2 pts) *w *Node*

j. (2 pts) w->next->next *Node* *

k. (2 pts) w->next *Node* *

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt cherry grape fig guava
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of argv[2][2]? a

c. (2 pts) What is the value of argv[1][1]? h

d. (2 pts) What is the value of argv[0][5]? I

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 001 100 101 from base 2 to base 8 145

   b. (2 pts) Convert 74 from base 8 to binary 111 100

   c. (2 pts) Convert 243 from decimal to binary 1111 0011

   d. (2 pts) Convert 0100 1110 1110 1001 from base 2 to hexadecimal 4ee9

   e. (2 pts) Convert 800f from hexadecimal to base 2 1000 0000 0000 1111

   f. (2 pts) Convert 1001 1001 from base 2 to decimal 153

   g. (2 pts) Convert 5f04 from hexadecimal to base 2 0101 1111 0000 0100

   h. (2 pts) Convert 20 from octal to base 2 010 000

   i. (2 pts) Convert 1010 1010 from binary to base 10 170

   j. (2 pts) Convert 110 001 101 from binary to octal 615

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double e;
  int f;
  Node g;
  char h;
  double *p;
  int *q;
  Node *r;
  char *s;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) r->next *Node **

b. (2 pts) s *char **

c. (2 pts) r->data *int*

d. (2 pts) *s *char*

e. (2 pts) r->next->next *Node **

f. (2 pts) argv[0] *char **

g. (2 pts) argc *int*

h. (2 pts) argv[1][2] *char*

i. (2 pts) &f *int **

j. (2 pts) f *int*

k. (2 pts) &s *char ***

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt apple banana guava mango
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of argv[2][5]? a

c. (2 pts) What is the value of argv[0][5]? I

d. (2 pts) What is the value of argv[1][3]? l

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 31 from octal to binary *011 001*

   b. (2 pts) Convert 1100 0011 0000 1000 from base 2 to base 16 *c308*

   c. (2 pts) Convert 12 from octal to binary *001 010*

   d. (2 pts) Convert 1010 1011 from binary to base 10 *171*

   e. (2 pts) Convert 1101 0110 0010 1101 from binary to base 16 *d62d*

   f. (2 pts) Convert 1100 1011 1001 1000 from binary to base 16 *cb98*

   g. (2 pts) Convert 0110 0001 from base 2 to decimal *97*

   h. (2 pts) Convert 52 from octal to base 2 *101 010*

   i. (2 pts) Convert 0100 1011 from binary to decimal *75*

   j. (2 pts) Convert f2e7 from hexadecimal to base 2 *1111 0010 1110 0111*

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int w;
  Node x;
  double y;
  char z;
  int *a;
  Node *b;
  double *c;
  char *d;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) y *double*

b. (2 pts) b->next *Node **

c. (2 pts) argc *int*

d. (2 pts) *d *char*

e. (2 pts) b->next->next *Node **

f. (2 pts) argv[0] *char **

g. (2 pts) &a *int ***

h. (2 pts) d *char **

i. (2 pts) b->data *int*

j. (2 pts) argv[1][2] *char*

k. (2 pts) &w *int **

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt date mango lime
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of argv[0][1]? /

c. (2 pts) What is the value of argv[2][2]? n

d. (2 pts) What is the value of argv[1][2]? t

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 1010 1001 from base 2 to decimal 169

   b. (2 pts) Convert 0111 1101 from binary to decimal 125

   c. (2 pts) Convert 101 111 from base 2 to octal 57

   d. (2 pts) Convert 15 from octal to binary 001 101

   e. (2 pts) Convert 56 from octal to base 2 101 110

   f. (2 pts) Convert 6575 from hexadecimal to base 2 0110 0101 0111 0101

   g. (2 pts) Convert 1011 from binary to decimal 11

   h. (2 pts) Convert 50 from base 10 to base 2 0011 0010

   i. (2 pts) Convert 21 from base 10 to base 2 0001 0101

   j. (2 pts) Convert 8ed1 from base 16 to base 2 1000 1110 1101 0001

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double r;
  Node s;
  int t;
  char w;
  double *x;
  Node *y;
  int *z;
  char *a;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) argv[0] *char **

b. (2 pts) y *Node **

c. (2 pts) &a *char ***

d. (2 pts) &w *char **

e. (2 pts) s *Node*

f. (2 pts) *x *double*

g. (2 pts) argv[1][2] *char*

h. (2 pts) y->data *int*

i. (2 pts) y->next->next *Node **

j. (2 pts) y->next *Node **

k. (2 pts) argc *int*

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt fig lemon
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of argv[1][1]? i

c. (2 pts) What is the value of argv[2][0]? l

d. (2 pts) What is the value of argv[0][1]? /

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 1 from decimal to binary 0001

   b. (2 pts) Convert 7 from octal to base 2 111

   c. (2 pts) Convert 1011 1101 from binary to base 10 189

   d. (2 pts) Convert f2a7 from base 16 to binary 1111 0010 1010 0111

   e. (2 pts) Convert 41 from decimal to binary 0010 1001

   f. (2 pts) Convert 31 from octal to base 2 011 001

   g. (2 pts) Convert 5cd6 from base 16 to binary 0101 1100 1101 0110

   h. (2 pts) Convert d89b from base 16 to base 2 1101 1000 1001 1011

   i. (2 pts) Convert 010 000 from base 2 to octal 20

   j. (2 pts) Convert 100 110 100 from base 2 to octal 464

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  Node d;
  double e;
  int f;
  char g;
  Node *h;
  double *p;
  int *q;
  char *r;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) d Node

b. (2 pts) argv[0] char *

c. (2 pts) h->next Node *

d. (2 pts) &p double **

e. (2 pts) argc int

f. (2 pts) h->next->next Node *

g. (2 pts) h->data int

h. (2 pts) *h Node

i. (2 pts) argv[1][2] char

j. (2 pts) &g char *

k. (2 pts) h Node *

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt fig lime
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of argv[0][1]? /

c. (2 pts) What is the value of argv[1][1]? i

d. (2 pts) What is the value of argv[2][3]? e

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 0100 0110 0100 1101 from base 2 to hexadecimal 464d

   b. (2 pts) Convert 110 110 100 from binary to base 8 664

   c. (2 pts) Convert 52 from base 8 to base 2 101 010

   d. (2 pts) Convert 7e91 from base 16 to base 2 0111 1110 1001 0001

   e. (2 pts) Convert 011 010 from binary to base 8 32

   f. (2 pts) Convert 1010 0010 from base 2 to base 16 a2

   g. (2 pts) Convert 0110 1100 0110 from binary to hexadecimal 6c6

   h. (2 pts) Convert 30 from base 8 to base 2 011 000

   i. (2 pts) Convert 1101 0010 from binary to base 10 210

   j. (2 pts) Convert 001 101 100 from base 2 to octal 154

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double a;
  int b;
  Node c;
  char d;
  double *e;
  int *f;
  Node *g;
  char *h;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) &e double **

b. (2 pts) g->next->next Node *

c. (2 pts) g->next Node *

d. (2 pts) c Node

e. (2 pts) argv[1][2] char

f. (2 pts) &c Node *

g. (2 pts) g->data int

h. (2 pts) h char *

i. (2 pts) argv[0] char *

j. (2 pts) *f int

k. (2 pts) argc int

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt date apple lime cherry
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of argv[1][1]? a

c. (2 pts) What is the value of argv[0][1]? /

d. (2 pts) What is the value of argv[2][3]? l

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 77af from hexadecimal to base 2 *0111 0111 1010 1111*

   b. (2 pts) Convert 1010 1011 from base 2 to base 10 *171*

   c. (2 pts) Convert 225 from base 10 to base 2 *1110 0001*

   d. (2 pts) Convert 1101 1010 1101 0011 from base 2 to base 16 *dad3*

   e. (2 pts) Convert 30 from octal to base 2 *011 000*

   f. (2 pts) Convert 001 100 110 from base 2 to octal *146*

   g. (2 pts) Convert 8f4 from base 16 to base 2 *1000 1111 0100*

   h. (2 pts) Convert 62 from octal to base 2 *110 010*

   i. (2 pts) Convert 011 101 000 from base 2 to octal *350*

   j. (2 pts) Convert 011 000 101 from base 2 to octal *305*

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int a;
  Node b;
  double c;
  char d;
  int *e;
  Node *f;
  double *g;
  char *h;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) f->next->next *Node **

b. (2 pts) f->next *Node **

c. (2 pts) argc *int*

d. (2 pts) *f *Node*

e. (2 pts) argv[1][2] *char*

f. (2 pts) a *int*

g. (2 pts) &a *int **

h. (2 pts) &f *Node ***

i. (2 pts) f->data *int*

j. (2 pts) g *double **

k. (2 pts) argv[0] *char **

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt kiwi cherry lime
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of argv[0][6]? t

c. (2 pts) What is the value of argv[2][1]? h

d. (2 pts) What is the value of argv[1][3]? i

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 57 from octal to binary 101 111

   b. (2 pts) Convert 101 from decimal to binary 0110 0101

   c. (2 pts) Convert 110 011 110 from binary to base 8 636

   d. (2 pts) Convert 0110 0110 1011 1101 from base 2 to hexadecimal 66bd

   e. (2 pts) Convert 010 001 101 from binary to octal 215

   f. (2 pts) Convert 110 011 001 from binary to base 8 631

   g. (2 pts) Convert b2e5 from hexadecimal to binary 1011 0010 1110 0101

   h. (2 pts) Convert 80 from decimal to base 2 0101 0000

   i. (2 pts) Convert 0011 1110 from binary to base 10 62

   j. (2 pts) Convert 111 111 100 from binary to octal 774

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  Node g;
  double h;
  int p;
  char q;
  Node *r;
  double *s;
  int *t;
  char *w;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) &q *char **

b. (2 pts) argv[0] *char **

c. (2 pts) argc *int*

d. (2 pts) &w *char ***

e. (2 pts) r->next->next *Node **

f. (2 pts) argv[1][2] *char*

g. (2 pts) *t *int*

h. (2 pts) r->next *Node **

i. (2 pts) p *int*

j. (2 pts) s *double **

k. (2 pts) r->data *int*

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt lemon fig
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of argv[2][0]? f

c. (2 pts) What is the value of argv[0][4]? n

d. (2 pts) What is the value of argv[1][0]? l

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 5 from octal to binary 101

   b. (2 pts) Convert 1000 from binary to decimal 8

   c. (2 pts) Convert 116 from base 10 to binary 0111 0100

   d. (2 pts) Convert 001 000 100 from base 2 to octal 104

   e. (2 pts) Convert 55 from octal to base 2 101 101

   f. (2 pts) Convert ce39 from hexadecimal to base 2 1100 1110 0011 1001

   g. (2 pts) Convert 001 001 from base 2 to base 8 11

   h. (2 pts) Convert 75 from octal to base 2 111 101

   i. (2 pts) Convert 0011 0000 from base 2 to decimal 48

   j. (2 pts) Convert 2 from base 8 to base 2 010

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  Node z;
  double a;
  int b;
  char c;
  Node *d;
  double *e;
  int *f;
  char *g;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) d->next->next Node *

b. (2 pts) argc int

c. (2 pts) d Node *

d. (2 pts) a double

e. (2 pts) &c char *

f. (2 pts) d->next Node *

g. (2 pts) argv[0] char *

h. (2 pts) d->data int

i. (2 pts) *f int

j. (2 pts) argv[1][2] char

k. (2 pts) &f int **

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt kiwi date
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of argv[1][0]? k

c. (2 pts) What is the value of argv[0][4]? n

d. (2 pts) What is the value of argv[2][0]? d

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 89 from decimal to binary 0101 1001

   b. (2 pts) Convert 1100 0010 1100 1000 from binary to base 16 c2c8

   c. (2 pts) Convert 011 000 101 from binary to octal 305

   d. (2 pts) Convert 101 011 100 from base 2 to base 8 534

   e. (2 pts) Convert 100 110 100 from binary to octal 464

   f. (2 pts) Convert 104 from decimal to base 2 0110 1000

   g. (2 pts) Convert 101 011 101 from binary to base 8 535

   h. (2 pts) Convert 37 from base 8 to base 2 011 111

   i. (2 pts) Convert 111 110 101 from binary to octal 765

   j. (2 pts) Convert 51 from octal to base 2 101 001

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int w;
  double x;
  Node y;
  char z;
  int *a;
  double *b;
  Node *c;
  char *d;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) w int

b. (2 pts) argc int

c. (2 pts) c->next->next Node *

d. (2 pts) d char *

e. (2 pts) c->next Node *

f. (2 pts) c->data int

g. (2 pts) &a int **

h. (2 pts) *d char

i. (2 pts) &y Node *

j. (2 pts) argv[1][2] char

k. (2 pts) argv[0] char *

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt fig date apple lemon
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of argv[0][3]? u

c. (2 pts) What is the value of argv[1][0]? f

d. (2 pts) What is the value of argv[2][1]? a

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 100 010 110 from base 2 to base 8 426

   b. (2 pts) Convert 944e from base 16 to binary 1001 0100 0100 1110

   c. (2 pts) Convert 98e4 from hexadecimal to base 2 1001 1000 1110 0100

   d. (2 pts) Convert 1010 from base 2 to decimal 10

   e. (2 pts) Convert 1111 0000 from binary to decimal 240

   f. (2 pts) Convert 46 from base 8 to binary 100 110

   g. (2 pts) Convert 1011 0000 1011 0111 from base 2 to hexadecimal b0b7

   h. (2 pts) Convert 231 from decimal to base 2 1110 0111

   i. (2 pts) Convert 100 111 001 from base 2 to octal 471

   j. (2 pts) Convert 64 from octal to binary 110 100

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  Node x;
  int y;
  double z;
  char a;
  Node *b;
  int *c;
  double *d;
  char *e;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) argc int

b. (2 pts) d double *

c. (2 pts) &d double **

d. (2 pts) y int

e. (2 pts) b->next->next Node *

f. (2 pts) &y int *

g. (2 pts) argv[1][2] char

h. (2 pts) argv[0] char *

i. (2 pts) b->data int

j. (2 pts) *e char

k. (2 pts) b->next Node *

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt kiwi lime guava
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of argv[2][0]? l

c. (2 pts) What is the value of argv[0][5]? I

d. (2 pts) What is the value of argv[1][0]? k

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert d00c from base 16 to base 2 1101 0000 0000 1100

   b. (2 pts) Convert 0100 1110 0111 0111 from base 2 to base 16 4e77

   c. (2 pts) Convert 86c5 from hexadecimal to binary 1000 0110 1100 0101

   d. (2 pts) Convert 150 from base 10 to base 2 1001 0110

   e. (2 pts) Convert d3d3 from hexadecimal to binary 1101 0011 1101 0011

   f. (2 pts) Convert 0011 0100 0101 1100 from binary to hexadecimal 345c

   g. (2 pts) Convert 0101 1010 1010 1000 from binary to base 16 5aa8

   h. (2 pts) Convert 111 from decimal to base 2 0110 1111

   i. (2 pts) Convert 011 001 101 from binary to base 8 315

   j. (2 pts) Convert 33 from base 8 to base 2 011 011

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double r;
  Node s;
  int t;
  char w;
  double *x;
  Node *y;
  int *z;
  char *a;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) argv[1][2] *char*

b. (2 pts) y *Node **

c. (2 pts) y->next *Node **

d. (2 pts) y->next->next *Node **

e. (2 pts) argv[0] *char **

f. (2 pts) &x *double ***

g. (2 pts) *x *double*

h. (2 pts) &r *double **

i. (2 pts) y->data *int*

j. (2 pts) argc *int*

k. (2 pts) w *char*

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt fig lime
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of argv[2][0]? l

c. (2 pts) What is the value of argv[1][0]? f

d. (2 pts) What is the value of argv[0][5]? I

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 2868 from base 16 to base 2 *0010 1000 0110 1000*

   b. (2 pts) Convert 241 from base 10 to binary *1111 0001*

   c. (2 pts) Convert 44 from decimal to base 2 *0010 1100*

   d. (2 pts) Convert 24 from octal to base 2 *010 100*

   e. (2 pts) Convert 21 from base 8 to base 2 *010 001*

   f. (2 pts) Convert 001 101 011 from binary to octal *153*

   g. (2 pts) Convert 1010 1100 from base 2 to decimal *172*

   h. (2 pts) Convert 5 from octal to base 2 *101*

   i. (2 pts) Convert 010 110 010 from base 2 to base 8 *262*

   j. (2 pts) Convert 0111 1001 1011 1000 from binary to hexadecimal *79b8*

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double d;
  Node e;
  int f;
  char g;
  double *h;
  Node *p;
  int *q;
  char *r;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) p->next *Node **

b. (2 pts) *r *char*

c. (2 pts) &r *char ***

d. (2 pts) argv[0] *char **

e. (2 pts) f *int*

f. (2 pts) &d *double **

g. (2 pts) p->next->next *Node **

h. (2 pts) r *char **

i. (2 pts) argv[1][2] *char*

j. (2 pts) p->data *int*

k. (2 pts) argc *int*

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt banana lime
```

a. (2 pts) What is the value of `argc` in this case? *3*

b. (2 pts) What is the value of argv[1][2]? *n*

c. (2 pts) What is the value of argv[2][0]? *l*

d. (2 pts) What is the value of argv[0][6]? *t*

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 011 011 010 from base 2 to octal 332

   b. (2 pts) Convert 52 from octal to base 2 101 010

   c. (2 pts) Convert 1a07 from hexadecimal to base 2 0001 1010 0000 0111

   d. (2 pts) Convert 1101 1101 1110 0011 from binary to hexadecimal dde3

   e. (2 pts) Convert 0010 0111 from base 2 to decimal 39

   f. (2 pts) Convert 63 from octal to base 2 110 011

   g. (2 pts) Convert 0101 0110 from binary to decimal 86

   h. (2 pts) Convert 157 from decimal to base 2 1001 1101

   i. (2 pts) Convert 10 from octal to binary 001 000

   j. (2 pts) Convert 0001 0101 1010 0011 from base 2 to hexadecimal 15a3

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double q;
  int r;
  Node s;
  char t;
  double *w;
  int *x;
  Node *y;
  char *z;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) y->data *int*

b. (2 pts) r *int*

c. (2 pts) *x *int*

d. (2 pts) &t *char **

e. (2 pts) argc *int*

f. (2 pts) argv[1][2] *char*

g. (2 pts) &x *int***

h. (2 pts) y->next->next *Node **

i. (2 pts) argv[0] *char **

j. (2 pts) y *Node **

k. (2 pts) y->next *Node **

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt banana date mango fig
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of argv[0][3]? u

c. (2 pts) What is the value of argv[2][2]? t

d. (2 pts) What is the value of argv[1][1]? a

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 1001 1110 1010 1010 from binary to hexadecimal 9eaa

   b. (2 pts) Convert 0111 1100 1111 0001 from base 2 to hexadecimal 7cf1

   c. (2 pts) Convert 010 100 000 from binary to octal 240

   d. (2 pts) Convert 0011 1010 from base 2 to decimal 58

   e. (2 pts) Convert 125 from base 10 to binary 0111 1101

   f. (2 pts) Convert 1 from base 10 to binary 0001

   g. (2 pts) Convert 010 110 000 from base 2 to base 8 260

   h. (2 pts) Convert 6 from base 10 to base 2 0110

   i. (2 pts) Convert 1100 0100 from base 2 to base 10 196

   j. (2 pts) Convert 20 from base 8 to base 2 010 000

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int r;
  Node s;
  double t;
  char w;
  int *x;
  Node *y;
  double *z;
  char *a;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) &s *Node **

b. (2 pts) *x *int*

c. (2 pts) &z *double***

d. (2 pts) y->data *int*

e. (2 pts) argv[1][2] *char*

f. (2 pts) argc *int*

g. (2 pts) argv[0] *char **

h. (2 pts) t *double*

i. (2 pts) y *Node **

j. (2 pts) y->next *Node **

k. (2 pts) y->next->next *Node **

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt apple cherry grape
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of argv[2][5]? y

c. (2 pts) What is the value of argv[0][3]? u

d. (2 pts) What is the value of argv[1][0]? a

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 1110 0011 from base 2 to base 10 227

   b. (2 pts) Convert 0011 0111 from base 2 to base 10 55

   c. (2 pts) Convert 0011 1110 0100 0110 from base 2 to hexadecimal 3e46

   d. (2 pts) Convert 61 from octal to binary 110 001

   e. (2 pts) Convert 0110 0001 0000 0000 from base 2 to base 16 6100

   f. (2 pts) Convert 1001 1011 from base 2 to decimal 155

   g. (2 pts) Convert 100 from binary to octal 4

   h. (2 pts) Convert 142 from base 10 to base 2 1000 1110

   i. (2 pts) Convert 100 011 110 from binary to octal 436

   j. (2 pts) Convert 1101 1101 1100 1110 from binary to base 16 ddce

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double h;
  Node p;
  int q;
  char r;
  double *s;
  Node *t;
  int *w;
  char *x;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) argv[0] *char **

b. (2 pts) *w *int*

c. (2 pts) argv[1][2] *char*

d. (2 pts) s *double **

e. (2 pts) t->data *int*

f. (2 pts) t->next->next *Node **

g. (2 pts) &t *Node ***

h. (2 pts) argc *int*

i. (2 pts) t->next *Node **

j. (2 pts) h *double*

k. (2 pts) &h *double **

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt grape mango
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of argv[2][3]? g

c. (2 pts) What is the value of argv[1][0]? g

d. (2 pts) What is the value of argv[0][1]? /

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 1001 1111 from binary to decimal 159

   b. (2 pts) Convert 100 101 001 from base 2 to octal 451

   c. (2 pts) Convert 279c from base 16 to base 2 0010 0111 1001 1100

   d. (2 pts) Convert 4 from base 10 to base 2 0100

   e. (2 pts) Convert 104 from decimal to binary 0110 1000

   f. (2 pts) Convert 011 000 011 from base 2 to base 8 303

   g. (2 pts) Convert 0001 0110 from binary to decimal 22

   h. (2 pts) Convert 17 from base 8 to binary 001 111

   i. (2 pts) Convert 62 from base 10 to base 2 0011 1110

   j. (2 pts) Convert 33 from octal to binary 011 011

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  Node x;
  double y;
  int z;
  char a;
  Node *b;
  double *c;
  int *d;
  char *e;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) &d int **

b. (2 pts) d int *

c. (2 pts) argv[1][2] char

d. (2 pts) &x Node *

e. (2 pts) argv[0] char *

f. (2 pts) b->next Node *

g. (2 pts) argc int

h. (2 pts) z int

i. (2 pts) b->data int

j. (2 pts) *d int

k. (2 pts) b->next->next Node *

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt cherry date
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of argv[1][4]? r

c. (2 pts) What is the value of argv[2][0]? d

d. (2 pts) What is the value of argv[0][1]? /

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert e4db from hexadecimal to binary 1110 0100 1101 1011

   b. (2 pts) Convert 4ef6 from hexadecimal to base 2 0100 1110 1111 0110

   c. (2 pts) Convert 0001 0101 from binary to base 10 21

   d. (2 pts) Convert 100 100 000 from base 2 to octal 440

   e. (2 pts) Convert 0100 1011 0101 1001 from binary to base 16 4b59

   f. (2 pts) Convert 76 from octal to base 2 111 110

   g. (2 pts) Convert 1011 1111 from base 2 to base 10 191

   h. (2 pts) Convert 198 from base 10 to binary 1100 0110

   i. (2 pts) Convert 8 from base 10 to binary 1000

   j. (2 pts) Convert 2 from base 8 to binary 010

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int r;
  double s;
  Node t;
  char w;
  int *x;
  double *y;
  Node *z;
  char *a;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) s *double*

b. (2 pts) z->next *Node **

c. (2 pts) y *double **

d. (2 pts) argv[1][2] *char*

e. (2 pts) argv[0] *char **

f. (2 pts) argc *int*

g. (2 pts) z->next->next *Node **

h. (2 pts) z->data *int*

i. (2 pts) &x *int ***

j. (2 pts) *a *char*

k. (2 pts) &w *char **

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt banana date kiwi mango
```

a. (2 pts) What is the value of `argc` in this case? *5*

b. (2 pts) What is the value of argv[0][2]? *r*

c. (2 pts) What is the value of argv[1][1]? *a*

d. (2 pts) What is the value of argv[2][2]? *t*

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 22 from base 10 to base 2 0001 0110

   b. (2 pts) Convert 32 from decimal to binary 0010 0000

   c. (2 pts) Convert 0100 1011 1101 1011 from binary to base 16 4bdb

   d. (2 pts) Convert 1110 1100 from base 2 to decimal 236

   e. (2 pts) Convert 50 from octal to base 2 101 000

   f. (2 pts) Convert 45 from base 10 to binary 0010 1101

   g. (2 pts) Convert 1100 0010 from binary to decimal 194

   h. (2 pts) Convert 47 from decimal to binary 0010 1111

   i. (2 pts) Convert 170 from base 10 to base 2 1010 1010

   j. (2 pts) Convert 001 101 000 from base 2 to octal 150

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double s;
  int t;
  Node w;
  char x;
  double *y;
  int *z;
  Node *a;
  char *b;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) &w Node *

b. (2 pts) argc int

c. (2 pts) argv[0] char *

d. (2 pts) a->data int

e. (2 pts) a->next->next Node *

f. (2 pts) &a Node **

g. (2 pts) w Node

h. (2 pts) a->next Node *

i. (2 pts) argv[1][2] char

j. (2 pts) y double *

k. (2 pts) *b char

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt guava grape fig kiwi
```

a. (2 pts) What is the value of `argc` in this case? *5*

b. (2 pts) What is the value of argv[0][2]? *r*

c. (2 pts) What is the value of argv[2][4]? *e*

d. (2 pts) What is the value of argv[1][3]? *v*

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 0101 1011 0001 1101 from base 2 to hexadecimal 5b1d

   b. (2 pts) Convert 66 from base 8 to base 2 110 110

   c. (2 pts) Convert 0011 1001 1011 1100 from base 2 to base 16 39bc

   d. (2 pts) Convert 120 from base 10 to binary 0111 1000

   e. (2 pts) Convert 1000 0100 from binary to base 10 132

   f. (2 pts) Convert 1100 0111 from base 2 to decimal 199

   g. (2 pts) Convert 0110 1100 from base 2 to decimal 108

   h. (2 pts) Convert b723 from base 16 to binary 1011 0111 0010 0011

   i. (2 pts) Convert 748e from hexadecimal to binary 0111 0100 1000 1110

   j. (2 pts) Convert 110 100 000 from base 2 to octal 640

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  Node a;
  int b;
  double c;
  char d;
  Node *e;
  int *f;
  double *g;
  char *h;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) e->next->next Node *

b. (2 pts) h char *

c. (2 pts) a Node

d. (2 pts) e->next Node *

e. (2 pts) argc int

f. (2 pts) *e Node

g. (2 pts) &b int *

h. (2 pts) e->data int

i. (2 pts) argv[0] char *

j. (2 pts) &e Node **

k. (2 pts) argv[1][2] char

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt banana cherry lemon
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of argv[1][4]? n

c. (2 pts) What is the value of argv[2][1]? h

d. (2 pts) What is the value of argv[0][3]? u

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 1011 0011 0111 1001 from base 2 to hexadecimal b379

   b. (2 pts) Convert 7d70 from base 16 to binary 0111 1101 0111 0000

   c. (2 pts) Convert 67 from octal to binary 110 111

   d. (2 pts) Convert 15 from base 8 to base 2 001 101

   e. (2 pts) Convert f53b from hexadecimal to binary 1111 0101 0011 1011

   f. (2 pts) Convert 200 from base 10 to binary 1100 1000

   g. (2 pts) Convert bdcc from hexadecimal to base 2 1011 1101 1100 1100

   h. (2 pts) Convert 010 111 011 from base 2 to octal 273

   i. (2 pts) Convert 0110 0111 from binary to decimal 103

   j. (2 pts) Convert 66 from base 8 to base 2 110 110

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double r;
  Node s;
  int t;
  char w;
  double *x;
  Node *y;
  int *z;
  char *a;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) y->data *int*

b. (2 pts) argc *int*

c. (2 pts) argv[0] *char **

d. (2 pts) y->next->next *Node **

e. (2 pts) r *double*

f. (2 pts) y->next *Node **

g. (2 pts) *x *double*

h. (2 pts) &s *Node **

i. (2 pts) y *Node **

j. (2 pts) argv[1][2] *char*

k. (2 pts) &a *char ***

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt cherry guava
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of argv[1][1]? h

c. (2 pts) What is the value of argv[2][0]? g

d. (2 pts) What is the value of argv[0][4]? n

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 1111 1000 from base 2 to decimal 248

   b. (2 pts) Convert 0011 0111 1001 1001 from base 2 to base 16 3799

   c. (2 pts) Convert 1100 1100 1111 1110 from binary to hexadecimal ccfe

   d. (2 pts) Convert 1011 1111 1110 1110 from binary to hexadecimal bfee

   e. (2 pts) Convert 66 from octal to base 2 110 110

   f. (2 pts) Convert 98 from decimal to base 2 0110 0010

   g. (2 pts) Convert 67bc from base 16 to binary 0110 0111 1011 1100

   h. (2 pts) Convert 229 from base 10 to binary 1110 0101

   i. (2 pts) Convert 49 from base 10 to base 2 0011 0001

   j. (2 pts) Convert 0111 0111 1010 1001 from binary to base 16 77a9

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double h;
  int p;
  Node q;
  char r;
  double *s;
  int *t;
  Node *w;
  char *x;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) q *Node*

b. (2 pts) w->next->next *Node **

c. (2 pts) s *double **

d. (2 pts) *t *int*

e. (2 pts) argc *int*

f. (2 pts) &t *int ***

g. (2 pts) &h *double **

h. (2 pts) argv[1][2] *char*

i. (2 pts) w->data *int*

j. (2 pts) w->next *Node **

k. (2 pts) argv[0] *char **

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt grape fig lime guava
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of argv[0][1]? /

c. (2 pts) What is the value of argv[2][1]? i

d. (2 pts) What is the value of argv[1][1]? r

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 12 from octal to base 2 001 010

   b. (2 pts) Convert 91e from hexadecimal to binary 1001 0001 1110

   c. (2 pts) Convert 0011 from base 2 to base 10 3

   d. (2 pts) Convert 0001 1100 from base 2 to base 10 28

   e. (2 pts) Convert 001 011 101 from binary to base 8 135

   f. (2 pts) Convert 1001 0101 0011 1101 from binary to hexadecimal 953d

   g. (2 pts) Convert 0110 1001 from binary to base 10 105

   h. (2 pts) Convert 4e34 from hexadecimal to binary 0100 1110 0011 0100

   i. (2 pts) Convert 1101 0010 from binary to base 10 210

   j. (2 pts) Convert 50 from base 8 to base 2 101 000

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int h;
  double p;
  Node q;
  char r;
  int *s;
  double *t;
  Node *w;
  char *x;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) w->data *int*

b. (2 pts) &w *Node **

c. (2 pts) argv[1][2] *char*

d. (2 pts) argv[0] *char **

e. (2 pts) w->next *Node **

f. (2 pts) r *char*

g. (2 pts) *t *double*

h. (2 pts) s *int **

i. (2 pts) &q *Node **

j. (2 pts) w->next->next *Node **

k. (2 pts) argc *int*

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt fig guava date grape
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of argv[2][1]? u

c. (2 pts) What is the value of argv[0][3]? u

d. (2 pts) What is the value of argv[1][2]? g

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 110 from base 10 to base 2 *0110 1110*

   b. (2 pts) Convert 110 000 110 from base 2 to base 8 *606*

   c. (2 pts) Convert 74 from base 8 to binary *111 100*

   d. (2 pts) Convert 52 from base 8 to binary *101 010*

   e. (2 pts) Convert 0001 0001 from binary to decimal *17*

   f. (2 pts) Convert 2f1a from base 16 to binary *0010 1111 0001 1010*

   g. (2 pts) Convert 0001 0011 from base 2 to base 10 *19*

   h. (2 pts) Convert 65 from octal to binary *110 101*

   i. (2 pts) Convert 156 from base 10 to binary *1001 1100*

   j. (2 pts) Convert 17 from octal to base 2 *001 111*

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  Node e;
  int f;
  double g;
  char h;
  Node *p;
  int *q;
  double *r;
  char *s;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) *r *double*

b. (2 pts) p->data *int*

c. (2 pts) s *char **

d. (2 pts) &q *int ***

e. (2 pts) argc *int*

f. (2 pts) p->next->next *Node **

g. (2 pts) p->next *Node **

h. (2 pts) argv[0] *char **

i. (2 pts) &f *int **

j. (2 pts) argv[1][2] *char*

k. (2 pts) f *int*

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt lime fig cherry
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of argv[2][0]? f

c. (2 pts) What is the value of argv[1][3]? e

d. (2 pts) What is the value of argv[0][0]? .

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 198 from decimal to base 2 1100 0110

   b. (2 pts) Convert 31 from base 8 to binary 011 001

   c. (2 pts) Convert 100 101 101 from binary to base 8 455

   d. (2 pts) Convert 63ad from hexadecimal to binary 0110 0011 1010 1101

   e. (2 pts) Convert 100 000 100 from binary to base 8 404

   f. (2 pts) Convert 0011 0000 from base 2 to decimal 48

   g. (2 pts) Convert 0110 0101 1000 1110 from base 2 to hexadecimal 658e

   h. (2 pts) Convert 124 from decimal to binary 0111 1100

   i. (2 pts) Convert 143 from decimal to binary 1000 1111

   j. (2 pts) Convert 0100 1011 0110 1000 from binary to hexadecimal 4b68

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  Node x;
  double y;
  int z;
  char a;
  Node *b;
  double *c;
  int *d;
  char *e;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) &e *char* **

b. (2 pts) &y *double* *

c. (2 pts) argv[0] *char* *

d. (2 pts) y *double*

e. (2 pts) b->data *int*

f. (2 pts) argv[1][2] *char*

g. (2 pts) b->next *Node* *

h. (2 pts) b->next->next *Node* *

i. (2 pts) *d *int*

j. (2 pts) c *double* *

k. (2 pts) argc *int*

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt grape apple
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of argv[1][3]? p

c. (2 pts) What is the value of argv[0][0]? .

d. (2 pts) What is the value of argv[2][3]? l

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 2 from base 8 to binary 010

   b. (2 pts) Convert 001 000 000 from base 2 to octal 100

   c. (2 pts) Convert 1000 0100 1000 0000 from base 2 to base 16 8480

   d. (2 pts) Convert ef97 from hexadecimal to base 2 1110 1111 1001 0111

   e. (2 pts) Convert 0110 0101 from binary to decimal 101

   f. (2 pts) Convert 110 010 100 from base 2 to octal 624

   g. (2 pts) Convert 1111 0111 1111 from binary to base 16 f7f

   h. (2 pts) Convert 4 from decimal to binary 0100

   i. (2 pts) Convert 0101 1001 from base 2 to base 10 89

   j. (2 pts) Convert 1110 0111 0101 0011 from base 2 to hexadecimal e753

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double d;
  Node e;
  int f;
  char g;
  double *h;
  Node *p;
  int *q;
  char *r;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) p->next->next Node *

b. (2 pts) *r char

c. (2 pts) &d double *

d. (2 pts) argc int

e. (2 pts) argv[0] char *

f. (2 pts) g char

g. (2 pts) argv[1][2] char

h. (2 pts) h double *

i. (2 pts) p->data int

j. (2 pts) &q int **

k. (2 pts) p->next Node *

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt date apple
```

a. (2 pts) What is the value of `argc` in this case? *3*

b. (2 pts) What is the value of argv[0][6]? *t*

c. (2 pts) What is the value of argv[1][0]? *d*

d. (2 pts) What is the value of argv[2][2]? *p*

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 3d38 from hexadecimal to binary 0011 1101 0011 1000

   b. (2 pts) Convert 1111 0001 from binary to decimal 241

   c. (2 pts) Convert 186 from base 10 to binary 1011 1010

   d. (2 pts) Convert 0100 1011 1101 1000 from binary to hexadecimal 4bd8

   e. (2 pts) Convert 187 from decimal to binary 1011 1011

   f. (2 pts) Convert fcb0 from hexadecimal to base 2 1111 1100 1011 0000

   g. (2 pts) Convert 0001 0001 1010 1101 from base 2 to hexadecimal 11ad

   h. (2 pts) Convert 6cf6 from base 16 to binary 0110 1100 1111 0110

   i. (2 pts) Convert fb2f from hexadecimal to binary 1111 1011 0010 1111

   j. (2 pts) Convert 0001 0011 1001 0100 from binary to base 16 1394

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int e;
  double f;
  Node g;
  char h;
  int *p;
  double *q;
  Node *r;
  char *s;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) argc *int*

b. (2 pts) &h *char **

c. (2 pts) s *char **

d. (2 pts) r->next->next *Node **

e. (2 pts) &s *char ***

f. (2 pts) argv[0] *char **

g. (2 pts) r->next *Node **

h. (2 pts) *s *char*

i. (2 pts) argv[1][2] *char*

j. (2 pts) e *int*

k. (2 pts) r->data *int*

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt guava grape apple fig
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of argv[2][0]? g

c. (2 pts) What is the value of argv[0][1]? /

d. (2 pts) What is the value of argv[1][2]? a

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 100 000 100 from binary to base 8 404

   b. (2 pts) Convert 171 from decimal to base 2 1010 1011

   c. (2 pts) Convert 168 from decimal to binary 1010 1000

   d. (2 pts) Convert 1101 0111 1100 0011 from binary to base 16 d7c3

   e. (2 pts) Convert 1001 1111 0001 1101 from binary to hexadecimal 9f1d

   f. (2 pts) Convert 150 from base 10 to binary 1001 0110

   g. (2 pts) Convert 1011 1011 1001 1110 from binary to hexadecimal bb9e

   h. (2 pts) Convert 75 from octal to binary 111 101

   i. (2 pts) Convert 1100 0101 from base 2 to base 10 197

   j. (2 pts) Convert 1010 1111 0111 1110 from base 2 to hexadecimal af7e

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int a;
  Node b;
  double c;
  char d;
  int *e;
  Node *f;
  double *g;
  char *h;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) *e int

b. (2 pts) &c double *

c. (2 pts) f->next Node *

d. (2 pts) argv[1][2] char

e. (2 pts) &f Node **

f. (2 pts) argc int

g. (2 pts) f->next->next Node *

h. (2 pts) c double

i. (2 pts) argv[0] char *

j. (2 pts) f->data int

k. (2 pts) g double *

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt lime guava kiwi
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of argv[0][4]? n

c. (2 pts) What is the value of argv[2][3]? v

d. (2 pts) What is the value of argv[1][0]? l

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 110 110 100 from binary to base 8 664

   b. (2 pts) Convert 010 011 101 from binary to base 8 235

   c. (2 pts) Convert 4e21 from base 16 to base 2 0100 1110 0010 0001

   d. (2 pts) Convert 100 100 110 from binary to octal 446

   e. (2 pts) Convert 011 111 from binary to base 8 37

   f. (2 pts) Convert 1001 0111 1101 1110 from base 2 to hexadecimal 97de

   g. (2 pts) Convert 011 010 from base 2 to octal 32

   h. (2 pts) Convert 155 from decimal to base 2 1001 1011

   i. (2 pts) Convert 1011 0111 from binary to decimal 183

   j. (2 pts) Convert bb12 from base 16 to base 2 1011 1011 0001 0010

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int r;
  double s;
  Node t;
  char w;
  int *x;
  double *y;
  Node *z;
  char *a;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) *x int

b. (2 pts) z->data int

c. (2 pts) t Node

d. (2 pts) y double *

e. (2 pts) argc int

f. (2 pts) &t Node *

g. (2 pts) &x int **

h. (2 pts) z->next Node *

i. (2 pts) z->next->next Node *

j. (2 pts) argv[0] char *

k. (2 pts) argv[1][2] char

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt mango lime fig banana
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of argv[0][6]? t

c. (2 pts) What is the value of argv[1][2]? n

d. (2 pts) What is the value of argv[2][1]? i

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 31 from decimal to base 2 *0001 1111*

   b. (2 pts) Convert 8df from base 16 to binary *1000 1101 1111*

   c. (2 pts) Convert 0011 1100 from binary to decimal *60*

   d. (2 pts) Convert 111 110 from binary to octal *76*

   e. (2 pts) Convert 242 from decimal to binary *1111 0010*

   f. (2 pts) Convert 31bb from base 16 to binary *0011 0001 1011 1011*

   g. (2 pts) Convert 1011 0111 from binary to decimal *183*

   h. (2 pts) Convert 22c8 from hexadecimal to base 2 *0010 0010 1100 1000*

   i. (2 pts) Convert 100 000 011 from base 2 to base 8 *403*

   j. (2 pts) Convert 56fc from base 16 to base 2 *0101 0110 1111 1100*

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int h;
  double p;
  Node q;
  char r;
  int *s;
  double *t;
  Node *w;
  char *x;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) h *int*

b. (2 pts) argv[0] *char **

c. (2 pts) argv[1][2] *char*

d. (2 pts) s *int **

e. (2 pts) &p *double **

f. (2 pts) &w *Node ***

g. (2 pts) w->data *int*

h. (2 pts) *t *double*

i. (2 pts) w->next *Node **

j. (2 pts) argc *int*

k. (2 pts) w->next->next *Node **

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt mango grape guava kiwi
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of argv[1][2]? n

c. (2 pts) What is the value of argv[0][0]? .

d. (2 pts) What is the value of argv[2][0]? g

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 0101 0000 from base 2 to decimal 80

   b. (2 pts) Convert 218 from decimal to base 2 1101 1010

   c. (2 pts) Convert 34 from octal to base 2 011 100

   d. (2 pts) Convert 0111 1011 from binary to decimal 123

   e. (2 pts) Convert 48ff from hexadecimal to binary 0100 1000 1111 1111

   f. (2 pts) Convert 31 from octal to base 2 011 001

   g. (2 pts) Convert 1011 1001 0111 0000 from base 2 to base 16 b970

   h. (2 pts) Convert 42 from base 8 to binary 100 010

   i. (2 pts) Convert 35 from decimal to binary 0010 0011

   j. (2 pts) Convert 1000 0011 0011 1101 from base 2 to hexadecimal 833d

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  Node a;
  int b;
  double c;
  char d;
  Node *e;
  int *f;
  double *g;
  char *h;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) e->next->next *Node **

b. (2 pts) argv[1][2] *char*

c. (2 pts) e->data *int*

d. (2 pts) argc *int*

e. (2 pts) *f *int*

f. (2 pts) e->next *Node **

g. (2 pts) &e *Node ***

h. (2 pts) argv[0] *char **

i. (2 pts) h *char **

j. (2 pts) &d *char **

k. (2 pts) b *int*

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt cherry lime kiwi
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of argv[2][3]? e

c. (2 pts) What is the value of argv[0][6]? t

d. (2 pts) What is the value of argv[1][5]? y

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 9594 from base 16 to binary 1001 0101 1001 0100

   b. (2 pts) Convert 45 from octal to binary 100 101

   c. (2 pts) Convert 6040 from hexadecimal to base 2 0110 0000 0100 0000

   d. (2 pts) Convert 7 from decimal to base 2 0111

   e. (2 pts) Convert 13 from octal to base 2 001 011

   f. (2 pts) Convert 1111 1110 0000 0001 from binary to base 16 fe01

   g. (2 pts) Convert 011 000 110 from binary to base 8 306

   h. (2 pts) Convert 100 110 from base 2 to base 8 46

   i. (2 pts) Convert 1110 1101 from base 2 to base 10 237

   j. (2 pts) Convert 1f28 from hexadecimal to binary 0001 1111 0010 1000

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double x;
  Node y;
  int z;
  char a;
  double *b;
  Node *c;
  int *d;
  char *e;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) c->next *Node **

b. (2 pts) argv[1][2] *char*

c. (2 pts) &z *int **

d. (2 pts) argc *int*

e. (2 pts) c->data *int*

f. (2 pts) d *int **

g. (2 pts) argv[0] *char **

h. (2 pts) c->next->next *Node **

i. (2 pts) x *double*

j. (2 pts) *d *int*

k. (2 pts) &d *int ***

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt date mango
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of argv[1][2]? t

c. (2 pts) What is the value of argv[2][4]? o

d. (2 pts) What is the value of argv[0][4]? n

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 1000 1001 1101 1010 from binary to base 16 *89da*

   b. (2 pts) Convert 124 from decimal to binary *0111 1100*

   c. (2 pts) Convert 110 000 011 from base 2 to base 8 *603*

   d. (2 pts) Convert 010 000 000 from base 2 to octal *200*

   e. (2 pts) Convert 1 from base 8 to binary *001*

   f. (2 pts) Convert 0011 1010 from binary to base 10 *58*

   g. (2 pts) Convert f327 from hexadecimal to base 2 *1111 0011 0010 0111*

   h. (2 pts) Convert 175 from decimal to binary *1010 1111*

   i. (2 pts) Convert 10 from base 8 to binary *001 000*

   j. (2 pts) Convert 52 from base 8 to binary *101 010*

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  Node r;
  double s;
  int t;
  char w;
  Node *x;
  double *y;
  int *z;
  char *a;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) x->next->next *Node **

b. (2 pts) argv[1][2] *char*

c. (2 pts) *y *double*

d. (2 pts) argc *int*

e. (2 pts) &t *int **

f. (2 pts) &z *int ***

g. (2 pts) x->data *int*

h. (2 pts) argv[0] *char **

i. (2 pts) y *double **

j. (2 pts) r *Node*

k. (2 pts) x->next *Node **

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt fig lemon
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of argv[1][0]? f

c. (2 pts) What is the value of argv[0][2]? r

d. (2 pts) What is the value of argv[2][0]? l

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 1100 1110 from binary to decimal 206

   b. (2 pts) Convert 369b from base 16 to base 2 0011 0110 1001 1011

   c. (2 pts) Convert 53 from base 8 to base 2 101 011

   d. (2 pts) Convert 110 011 000 from base 2 to base 8 630

   e. (2 pts) Convert 1110 1001 from base 2 to decimal 233

   f. (2 pts) Convert 110 101 001 from base 2 to base 8 651

   g. (2 pts) Convert 9d18 from base 16 to binary 1001 1101 0001 1000

   h. (2 pts) Convert 15 from octal to binary 001 101

   i. (2 pts) Convert 1110 1101 0001 0001 from binary to base 16 ed11

   j. (2 pts) Convert 0100 0011 1111 0011 from binary to hexadecimal 43f3

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int z;
  double a;
  Node b;
  char c;
  int *d;
  double *e;
  Node *f;
  char *g;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) argv[0] *char **

b. (2 pts) f->next->next *Node **

c. (2 pts) f->data *int*

d. (2 pts) argv[1][2] *char*

e. (2 pts) &a *double **

f. (2 pts) *g *char*

g. (2 pts) argc *int*

h. (2 pts) &d *int ***

i. (2 pts) f->next *Node **

j. (2 pts) d *int **

k. (2 pts) b *Node*

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt date lime kiwi mango
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of argv[0][5]? I

c. (2 pts) What is the value of argv[1][0]? d

d. (2 pts) What is the value of argv[2][3]? e

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 0 from base 8 to binary 000

   b. (2 pts) Convert 8 from base 10 to binary 1000

   c. (2 pts) Convert 229 from base 10 to binary 1110 0101

   d. (2 pts) Convert 0010 1000 from base 2 to base 10 40

   e. (2 pts) Convert 63 from base 10 to base 2 0011 1111

   f. (2 pts) Convert 7 from decimal to binary 0111

   g. (2 pts) Convert 9f46 from hexadecimal to base 2 1001 1111 0100 0110

   h. (2 pts) Convert 50 from base 8 to binary 101 000

   i. (2 pts) Convert 43 from octal to binary 100 011

   j. (2 pts) Convert 0111 0000 from base 2 to base 10 112

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double z;
  int a;
  Node b;
  char c;
  double *d;
  int *e;
  Node *f;
  char *g;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) argv[0] *char **

b. (2 pts) c *char*

c. (2 pts) &f *Node ***

d. (2 pts) argv[1][2] *char*

e. (2 pts) f->data *int*

f. (2 pts) &z *double **

g. (2 pts) f->next->next *Node **

h. (2 pts) argc *int*

i. (2 pts) *f *Node*

j. (2 pts) f->next *Node **

k. (2 pts) d *double **

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt cherry mango kiwi date
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of argv[2][4]? o

c. (2 pts) What is the value of argv[0][3]? u

d. (2 pts) What is the value of argv[1][0]? c

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 68 from decimal to base 2 *0100 0100*

   b. (2 pts) Convert 60 from octal to base 2 *110 000*

   c. (2 pts) Convert 211 from decimal to binary *1101 0011*

   d. (2 pts) Convert 180 from base 10 to binary *1011 0100*

   e. (2 pts) Convert 22e4 from base 16 to base 2 *0010 0010 1110 0100*

   f. (2 pts) Convert 161 from decimal to binary *1010 0001*

   g. (2 pts) Convert 4936 from base 16 to base 2 *0100 1001 0011 0110*

   h. (2 pts) Convert 40 from base 10 to binary *0010 1000*

   i. (2 pts) Convert 0101 1000 1110 0011 from base 2 to base 16 *58e3*

   j. (2 pts) Convert 3 from octal to base 2 *011*

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int w;
  Node x;
  double y;
  char z;
  int *a;
  Node *b;
  double *c;
  char *d;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) argv[1][2] *char*

b. (2 pts) y *double*

c. (2 pts) b->next->next *Node **

d. (2 pts) &a *int***

e. (2 pts) argv[0] *char **

f. (2 pts) b->data *int*

g. (2 pts) argc *int*

h. (2 pts) d *char **

i. (2 pts) &z *char **

j. (2 pts) b->next *Node **

k. (2 pts) *a *int*

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt apple cherry lime
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of argv[2][2]? e

c. (2 pts) What is the value of argv[1][2]? p

d. (2 pts) What is the value of argv[0][2]? r

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 157 from base 10 to base 2 1001 1101

   b. (2 pts) Convert 6516 from hexadecimal to binary 0110 0101 0001 0110

   c. (2 pts) Convert 0111 1001 from base 2 to base 10 121

   d. (2 pts) Convert 33 from octal to binary 011 011

   e. (2 pts) Convert 147 from base 10 to binary 1001 0011

   f. (2 pts) Convert a262 from hexadecimal to binary 1010 0010 0110 0010

   g. (2 pts) Convert 100 110 101 from binary to base 8 465

   h. (2 pts) Convert 1100 1110 1010 0110 from base 2 to hexadecimal cea6

   i. (2 pts) Convert 0100 1011 0101 0111 from base 2 to hexadecimal 4b57

   j. (2 pts) Convert 0001 0111 1011 0010 from binary to hexadecimal 17b2

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double g;
  Node h;
  int p;
  char q;
  double *r;
  Node *s;
  int *t;
  char *w;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) s *Node **

b. (2 pts) s->next *Node **

c. (2 pts) argc *int*

d. (2 pts) argv[0] *char **

e. (2 pts) &w *char ***

f. (2 pts) &q *char **

g. (2 pts) argv[1][2] *char*

h. (2 pts) h *Node*

i. (2 pts) s->next->next *Node **

j. (2 pts) *w *char*

k. (2 pts) s->data *int*

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt guava grape
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of argv[1][2]? a

c. (2 pts) What is the value of argv[2][3]? p

d. (2 pts) What is the value of argv[0][0]? .

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 1110 0010 0011 0111 from binary to base 16 e237

   b. (2 pts) Convert 0001 1111 0011 1110 from binary to base 16 1f3e

   c. (2 pts) Convert 0110 0111 from binary to decimal 103

   d. (2 pts) Convert 1111 1011 1101 1000 from binary to base 16 fbd8

   e. (2 pts) Convert 76a8 from base 16 to base 2 0111 0110 1010 1000

   f. (2 pts) Convert 60 from base 10 to base 2 0011 1100

   g. (2 pts) Convert 010 001 001 from base 2 to base 8 211

   h. (2 pts) Convert 25 from octal to binary 010 101

   i. (2 pts) Convert 5 from octal to base 2 101

   j. (2 pts) Convert 1011 0011 1001 1101 from base 2 to base 16 b39d

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int d;
  double e;
  Node f;
  char g;
  int *h;
  double *p;
  Node *q;
  char *r;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) h *int **

b. (2 pts) argv[1][2] *char*

c. (2 pts) argv[0] *char **

d. (2 pts) g *char*

e. (2 pts) &p *double ***

f. (2 pts) q->next->next *Node **

g. (2 pts) q->data *int*

h. (2 pts) q->next *Node **

i. (2 pts) argc *int*

j. (2 pts) *h *int*

k. (2 pts) &f *Node **

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt cherry mango guava banana
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of argv[0][4]? n

c. (2 pts) What is the value of argv[2][3]? g

d. (2 pts) What is the value of argv[1][5]? y

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 1399 from base 16 to binary 0001 0011 1001 1001

   b. (2 pts) Convert 74 from base 8 to binary 111 100

   c. (2 pts) Convert 47 from octal to base 2 100 111

   d. (2 pts) Convert 26 from octal to binary 010 110

   e. (2 pts) Convert 1100 1100 1100 0110 from base 2 to base 16 ccc6

   f. (2 pts) Convert 33 from base 8 to binary 011 011

   g. (2 pts) Convert 4708 from hexadecimal to binary 0100 0111 0000 1000

   h. (2 pts) Convert 1011 1111 from binary to decimal 191

   i. (2 pts) Convert 1011 0111 0010 1000 from base 2 to hexadecimal b728

   j. (2 pts) Convert 1101 1111 1101 1101 from binary to hexadecimal dfdd

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double t;
  int w;
  Node x;
  char y;
  double *z;
  int *a;
  Node *b;
  char *c;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) b->data *int*

b. (2 pts) b->next->next *Node **

c. (2 pts) &t *double **

d. (2 pts) *z *double*

e. (2 pts) argc *int*

f. (2 pts) argv[0] *char **

g. (2 pts) w *int*

h. (2 pts) b->next *Node **

i. (2 pts) argv[1][2] *char*

j. (2 pts) c *char **

k. (2 pts) &b *Node ***

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt mango lemon date fig
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of argv[1][3]? g

c. (2 pts) What is the value of argv[2][0]? l

d. (2 pts) What is the value of argv[0][6]? t

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 26 from base 8 to binary 010 110

   b. (2 pts) Convert 101 010 101 from base 2 to base 8 525

   c. (2 pts) Convert 8b4f from base 16 to base 2 1000 1011 0100 1111

   d. (2 pts) Convert 71 from octal to base 2 111 001

   e. (2 pts) Convert 101 100 000 from base 2 to octal 540

   f. (2 pts) Convert 1000 1000 0101 from binary to hexadecimal 885

   g. (2 pts) Convert 111 100 001 from base 2 to octal 741

   h. (2 pts) Convert 70 from base 10 to binary 0100 0110

   i. (2 pts) Convert 40 from octal to base 2 100 000

   j. (2 pts) Convert 0111 1011 1100 1000 from base 2 to hexadecimal 7bc8

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  Node q;
  int r;
  double s;
  char t;
  Node *w;
  int *x;
  double *y;
  char *z;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) argc int

b. (2 pts) &x int **

c. (2 pts) w->data int

d. (2 pts) w->next Node *

e. (2 pts) t char

f. (2 pts) y double *

g. (2 pts) argv[1][2] char

h. (2 pts) *x int

i. (2 pts) w->next->next Node *

j. (2 pts) &q Node *

k. (2 pts) argv[0] char *

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt banana lime apple
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of argv[2][3]? e

c. (2 pts) What is the value of argv[1][5]? a

d. (2 pts) What is the value of argv[0][3]? u

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 101 100 001 from base 2 to octal 541

   b. (2 pts) Convert 23 from octal to binary 010 011

   c. (2 pts) Convert 0011 0000 1011 0001 from binary to hexadecimal 30b1

   d. (2 pts) Convert 9f96 from hexadecimal to base 2 1001 1111 1001 0110

   e. (2 pts) Convert 32 from base 10 to base 2 0010 0000

   f. (2 pts) Convert 9 from decimal to binary 1001

   g. (2 pts) Convert 010 000 101 from binary to octal 205

   h. (2 pts) Convert 1110 1101 0110 1000 from base 2 to hexadecimal ed68

   i. (2 pts) Convert 73b3 from base 16 to binary 0111 0011 1011 0011

   j. (2 pts) Convert 875c from base 16 to base 2 1000 0111 0101 1100

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double c;
  Node d;
  int e;
  char f;
  double *g;
  Node *h;
  int *p;
  char *q;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) argv[1][2] *char*

b. (2 pts) &f *char **

c. (2 pts) h->next->next *Node **

d. (2 pts) argc *int*

e. (2 pts) h->next *Node **

f. (2 pts) *h *Node*

g. (2 pts) h->data *int*

h. (2 pts) &q *char ***

i. (2 pts) h *Node **

j. (2 pts) argv[0] *char **

k. (2 pts) e *int*

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt kiwi grape
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of argv[0][4]? n

c. (2 pts) What is the value of argv[1][0]? k

d. (2 pts) What is the value of argv[2][4]? e

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 245 from base 10 to base 2 1111 0101

   b. (2 pts) Convert 7 from decimal to binary 0111

   c. (2 pts) Convert 30 from decimal to binary 0001 1110

   d. (2 pts) Convert 2b80 from hexadecimal to binary 0010 1011 1000 0000

   e. (2 pts) Convert 0011 1101 0101 from base 2 to base 16 3d5

   f. (2 pts) Convert 1010 0011 from binary to base 10 163

   g. (2 pts) Convert 111 011 001 from base 2 to octal 731

   h. (2 pts) Convert 35 from base 8 to binary 011 101

   i. (2 pts) Convert 0011 1101 1100 1010 from binary to base 16 3dca

   j. (2 pts) Convert 2346 from base 16 to base 2 0010 0011 0100 0110

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double z;
  int a;
  Node b;
  char c;
  double *d;
  int *e;
  Node *f;
  char *g;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) argv[0] *char **

b. (2 pts) argv[1][2] *char*

c. (2 pts) &f *Node ***

d. (2 pts) d *double **

e. (2 pts) f->next->next *Node **

f. (2 pts) argc *int*

g. (2 pts) &b *Node **

h. (2 pts) a *int*

i. (2 pts) f->data *int*

j. (2 pts) f->next *Node **

k. (2 pts) *f *Node*

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt mango lemon cherry fig
```

a. (2 pts) What is the value of `argc` in this case? *5*

b. (2 pts) What is the value of argv[1][0]? *m*

c. (2 pts) What is the value of argv[0][0]? *.*

d. (2 pts) What is the value of argv[2][2]? *m*

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 0010 0111 from binary to base 10 39

   b. (2 pts) Convert 1101 1001 from base 2 to decimal 217

   c. (2 pts) Convert 010 101 001 from base 2 to octal 251

   d. (2 pts) Convert 1000 0111 1100 0010 from binary to hexadecimal 87c2

   e. (2 pts) Convert 26 from base 8 to binary 010 110

   f. (2 pts) Convert 214 from base 10 to base 2 1101 0110

   g. (2 pts) Convert 111 011 101 from binary to base 8 735

   h. (2 pts) Convert 1101 1110 from binary to decimal 222

   i. (2 pts) Convert 1101 1111 1000 0101 from base 2 to hexadecimal df85

   j. (2 pts) Convert 0100 1111 1000 0111 from base 2 to base 16 4f87

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int a;
  double b;
  Node c;
  char d;
  int *e;
  double *f;
  Node *g;
  char *h;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) g->data *int*

b. (2 pts) argv[1][2] *char*

c. (2 pts) argv[0] *char **

d. (2 pts) *g *Node*

e. (2 pts) h *char **

f. (2 pts) g->next *Node **

g. (2 pts) c *Node*

h. (2 pts) &b *double **

i. (2 pts) &h *char ***

j. (2 pts) g->next->next *Node **

k. (2 pts) argc *int*

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt kiwi cherry lemon banana
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of argv[0][2]? r

c. (2 pts) What is the value of argv[2][3]? r

d. (2 pts) What is the value of argv[1][1]? i

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 6bf5 from hexadecimal to binary 0110 1011 1111 0101

   b. (2 pts) Convert 1001 0011 1001 0000 from base 2 to base 16 9390

   c. (2 pts) Convert 0100 0010 1101 0001 from base 2 to hexadecimal 42d1

   d. (2 pts) Convert 19 from decimal to binary 0001 0011

   e. (2 pts) Convert 0011 1101 from base 2 to decimal 61

   f. (2 pts) Convert 6ff9 from hexadecimal to binary 0110 1111 1111 1001

   g. (2 pts) Convert 100 110 001 from base 2 to base 8 461

   h. (2 pts) Convert 101 from decimal to binary 0110 0101

   i. (2 pts) Convert 1010 1001 1001 1100 from binary to base 16 a99c

   j. (2 pts) Convert eb71 from hexadecimal to base 2 1110 1011 0111 0001

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int f;
  Node g;
  double h;
  char p;
  int *q;
  Node *r;
  double *s;
  char *t;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) &r Node **

b. (2 pts) r->data int

c. (2 pts) f int

d. (2 pts) argc int

e. (2 pts) s double *

f. (2 pts) *t char

g. (2 pts) r->next->next Node *

h. (2 pts) argv[0] char *

i. (2 pts) &f int *

j. (2 pts) argv[1][2] char

k. (2 pts) r->next Node *

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt cherry lime apple
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of argv[1][4]? r

c. (2 pts) What is the value of argv[2][2]? m

d. (2 pts) What is the value of argv[0][6]? t

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 1100 0100 0101 0010 from binary to hexadecimal c452

   b. (2 pts) Convert 001 101 100 from base 2 to base 8 154

   c. (2 pts) Convert 72 from octal to base 2 111 010

   d. (2 pts) Convert 110 011 110 from binary to base 8 636

   e. (2 pts) Convert adb7 from base 16 to base 2 1010 1101 1011 0111

   f. (2 pts) Convert 0111 0001 0100 1001 from binary to hexadecimal 7149

   g. (2 pts) Convert ea6e from hexadecimal to base 2 1110 1010 0110 1110

   h. (2 pts) Convert 3 from octal to base 2 011

   i. (2 pts) Convert 47 from base 8 to binary 100 111

   j. (2 pts) Convert 247 from decimal to binary 1111 0111

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int z;
  double a;
  Node b;
  char c;
  int *d;
  double *e;
  Node *f;
  char *g;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) argc *int*

b. (2 pts) &d *int* **

c. (2 pts) c *char*

d. (2 pts) f->next *Node* *

e. (2 pts) &z *int* *

f. (2 pts) f->data *int*

g. (2 pts) argv[1][2] *char*

h. (2 pts) argv[0] *char* *

i. (2 pts) f->next->next *Node* *

j. (2 pts) d *int* *

k. (2 pts) *g *char*

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt fig guava apple date
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of argv[0][2]? r

c. (2 pts) What is the value of argv[1][1]? i

d. (2 pts) What is the value of argv[2][2]? a

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 010 010 from binary to octal 22

   b. (2 pts) Convert f085 from hexadecimal to binary 1111 0000 1000 0101

   c. (2 pts) Convert 65 from base 8 to base 2 110 101

   d. (2 pts) Convert 010 110 110 from base 2 to octal 266

   e. (2 pts) Convert 44 from base 8 to binary 100 100

   f. (2 pts) Convert 1011 0010 0110 from binary to hexadecimal b26

   g. (2 pts) Convert 945f from base 16 to base 2 1001 0100 0101 1111

   h. (2 pts) Convert 1001 0011 from base 2 to base 10 147

   i. (2 pts) Convert 6627 from hexadecimal to base 2 0110 0110 0010 0111

   j. (2 pts) Convert 146 from decimal to binary 1001 0010

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int t;
  double w;
  Node x;
  char y;
  int *z;
  double *a;
  Node *b;
  char *c;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) b->next->next *Node **

b. (2 pts) *z *int*

c. (2 pts) argc *int*

d. (2 pts) b->next *Node **

e. (2 pts) x *Node*

f. (2 pts) argv[0] *char **

g. (2 pts) c *char **

h. (2 pts) &y *char **

i. (2 pts) b->data *int*

j. (2 pts) &b *Node ***

k. (2 pts) argv[1][2] *char*

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt banana grape mango fig
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of argv[1][1]? a

c. (2 pts) What is the value of argv[0][5]? I

d. (2 pts) What is the value of argv[2][0]? g

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 16 from base 8 to base 2 *001 110*

   b. (2 pts) Convert 194 from base 10 to base 2 *1100 0010*

   c. (2 pts) Convert 12 from base 10 to base 2 *1100*

   d. (2 pts) Convert 1011 0111 from binary to base 10 *183*

   e. (2 pts) Convert 111 001 110 from base 2 to octal *716*

   f. (2 pts) Convert 001 111 011 from base 2 to base 8 *173*

   g. (2 pts) Convert 968d from hexadecimal to binary *1001 0110 1000 1101*

   h. (2 pts) Convert 252 from base 10 to binary *1111 1100*

   i. (2 pts) Convert 7e1 from base 16 to binary *0111 1110 0001*

   j. (2 pts) Convert bf31 from hexadecimal to binary *1011 1111 0011 0001*

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double w;
  int x;
  Node y;
  char z;
  double *a;
  int *b;
  Node *c;
  char *d;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) d *char **

b. (2 pts) &a *double ***

c. (2 pts) argc *int*

d. (2 pts) &x *int **

e. (2 pts) c->next *Node **

f. (2 pts) c->data *int*

g. (2 pts) argv[0] *char **

h. (2 pts) *a *double*

i. (2 pts) c->next->next *Node **

j. (2 pts) z *char*

k. (2 pts) argv[1][2] *char*

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt mango date banana lemon
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of argv[0][0]? .

c. (2 pts) What is the value of argv[2][0]? d

d. (2 pts) What is the value of argv[1][0]? m

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 0111 1111 from binary to base 10 *127*

   b. (2 pts) Convert 37 from base 8 to binary *011 111*

   c. (2 pts) Convert 1111 1010 from binary to base 10 *250*

   d. (2 pts) Convert 67 from decimal to binary *0100 0011*

   e. (2 pts) Convert 1100 1010 from base 2 to base 10 *202*

   f. (2 pts) Convert 65 from base 8 to base 2 *110 101*

   g. (2 pts) Convert 407e from base 16 to binary *0100 0000 0111 1110*

   h. (2 pts) Convert 1000 0100 0111 1001 from binary to hexadecimal *8479*

   i. (2 pts) Convert 1101 0001 1111 1000 from binary to base 16 *d1f8*

   j. (2 pts) Convert 5b1b from base 16 to binary *0101 1011 0001 1011*

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  Node r;
  double s;
  int t;
  char w;
  Node *x;
  double *y;
  int *z;
  char *a;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) argv[0] *char **

b. (2 pts) &z *int ***

c. (2 pts) x->next *Node **

d. (2 pts) *y *double*

e. (2 pts) &s *double **

f. (2 pts) s *double*

g. (2 pts) argv[1][2] *char*

h. (2 pts) x->next->next *Node **

i. (2 pts) argc *int*

j. (2 pts) x->data *int*

k. (2 pts) y *double **

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt cherry mango
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of argv[1][0]? c

c. (2 pts) What is the value of argv[2][2]? n

d. (2 pts) What is the value of argv[0][3]? u

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 0011 1011 from binary to base 10 59

   b. (2 pts) Convert 1101 1001 1110 0110 from base 2 to hexadecimal d9e6

   c. (2 pts) Convert 70 from octal to base 2 111 000

   d. (2 pts) Convert 81a2 from hexadecimal to base 2 1000 0001 1010 0010

   e. (2 pts) Convert 110 100 010 from base 2 to octal 642

   f. (2 pts) Convert 1001 1101 0010 0000 from base 2 to hexadecimal 9d20

   g. (2 pts) Convert 541d from base 16 to base 2 0101 0100 0001 1101

   h. (2 pts) Convert 15 from base 8 to base 2 001 101

   i. (2 pts) Convert 100 000 011 from binary to base 8 403

   j. (2 pts) Convert 1110 1001 0110 0010 from base 2 to hexadecimal e962

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  Node a;
  double b;
  int c;
  char d;
  Node *e;
  double *f;
  int *g;
  char *h;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) argv[1][2] *char*

b. (2 pts) d *char*

c. (2 pts) &a *Node **

d. (2 pts) &h *char ***

e. (2 pts) e->next *Node **

f. (2 pts) argv[0] *char **

g. (2 pts) argc *int*

h. (2 pts) e->next->next *Node **

i. (2 pts) h *char **

j. (2 pts) *f *double*

k. (2 pts) e->data *int*

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt lemon grape
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of argv[0][3]? u

c. (2 pts) What is the value of argv[1][1]? e

d. (2 pts) What is the value of argv[2][0]? g

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 80c5 from base 16 to base 2 1000 0000 1100 0101

   b. (2 pts) Convert 1001 0100 from binary to decimal 148

   c. (2 pts) Convert d189 from base 16 to binary 1101 0001 1000 1001

   d. (2 pts) Convert d8c from hexadecimal to binary 1101 1000 1100

   e. (2 pts) Convert 180 from base 10 to binary 1011 0100

   f. (2 pts) Convert 0011 0110 1111 1101 from binary to hexadecimal 36fd

   g. (2 pts) Convert fe0e from hexadecimal to base 2 1111 1110 0000 1110

   h. (2 pts) Convert 188 from decimal to base 2 1011 1100

   i. (2 pts) Convert 0100 1011 from base 2 to decimal 75

   j. (2 pts) Convert 854c from hexadecimal to base 2 1000 0101 0100 1100

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double x;
  int y;
  Node z;
  char a;
  double *b;
  int *c;
  Node *d;
  char *e;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) argv[1][2] *char*

b. (2 pts) argv[0] *char **

c. (2 pts) d->next *Node **

d. (2 pts) &d *Node ***

e. (2 pts) *e *char*

f. (2 pts) d->data *int*

g. (2 pts) z *Node*

h. (2 pts) d->next->next *Node **

i. (2 pts) &a *char **

j. (2 pts) argc *int*

k. (2 pts) d *Node **

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt guava kiwi mango grape
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of argv[1][2]? a

c. (2 pts) What is the value of argv[0][2]? r

d. (2 pts) What is the value of argv[2][3]? i

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 178 from decimal to binary 1011 0010

   b. (2 pts) Convert 011 001 011 from binary to octal 313

   c. (2 pts) Convert 0111 from base 2 to base 10 7

   d. (2 pts) Convert 0110 1001 1100 1101 from binary to base 16 69cd

   e. (2 pts) Convert ae3 from base 16 to base 2 1010 1110 0011

   f. (2 pts) Convert 011 010 010 from base 2 to base 8 322

   g. (2 pts) Convert 0000 from base 2 to base 10 0

   h. (2 pts) Convert 37 from base 10 to base 2 0010 0101

   i. (2 pts) Convert 111 011 010 from base 2 to base 8 732

   j. (2 pts) Convert 1011 0001 1000 1101 from binary to hexadecimal b18d

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int g;
  Node h;
  double p;
  char q;
  int *r;
  Node *s;
  double *t;
  char *w;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) s->next->next Node *

b. (2 pts) q char

c. (2 pts) argv[0] char *

d. (2 pts) &p double *

e. (2 pts) argv[1][2] char

f. (2 pts) s->next Node *

g. (2 pts) argc int

h. (2 pts) s->data int

i. (2 pts) *t double

j. (2 pts) &w char **

k. (2 pts) s Node *

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt lime kiwi banana
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of argv[2][3]? i

c. (2 pts) What is the value of argv[0][6]? t

d. (2 pts) What is the value of argv[1][2]? m

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 1111 0111 0000 0110 from binary to hexadecimal f706

   b. (2 pts) Convert 1fbd from base 16 to base 2 0001 1111 1011 1101

   c. (2 pts) Convert 1111 0101 from base 2 to base 10 245

   d. (2 pts) Convert 245 from decimal to binary 1111 0101

   e. (2 pts) Convert 1110 1110 0010 1110 from base 2 to base 16 ee2e

   f. (2 pts) Convert 0 from base 8 to binary 000

   g. (2 pts) Convert aa2d from hexadecimal to base 2 1010 1010 0010 1101

   h. (2 pts) Convert 53 from base 8 to base 2 101 011

   i. (2 pts) Convert 55 from octal to binary 101 101

   j. (2 pts) Convert 0100 1101 0111 1000 from base 2 to hexadecimal 4d78

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  Node d;
  double e;
  int f;
  char g;
  Node *h;
  double *p;
  int *q;
  char *r;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) argc *int*

b. (2 pts) e *double*

c. (2 pts) h->next->next *Node **

d. (2 pts) &e *double **

e. (2 pts) h *Node **

f. (2 pts) h->data *int*

g. (2 pts) h->next *Node **

h. (2 pts) argv[1][2] *char*

i. (2 pts) &p *double ***

j. (2 pts) *h *Node*

k. (2 pts) argv[0] *char **

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt banana kiwi
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of argv[2][3]? i

c. (2 pts) What is the value of argv[1][3]? a

d. (2 pts) What is the value of argv[0][5]? I

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 23 from octal to binary 010 011

   b. (2 pts) Convert 1100 0010 from base 2 to base 10 194

   c. (2 pts) Convert 100 110 110 from base 2 to octal 466

   d. (2 pts) Convert 101 100 010 from binary to octal 542

   e. (2 pts) Convert 5ea7 from base 16 to binary 0101 1110 1010 0111

   f. (2 pts) Convert 0100 from base 2 to decimal 4

   g. (2 pts) Convert 1111 1011 1110 0000 from binary to hexadecimal fbe0

   h. (2 pts) Convert 0101 0011 from base 2 to decimal 83

   i. (2 pts) Convert 52 from octal to base 2 101 010

   j. (2 pts) Convert 590c from hexadecimal to base 2 0101 1001 0000 1100

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  Node w;
  double x;
  int y;
  char z;
  Node *a;
  double *b;
  int *c;
  char *d;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) argc int

b. (2 pts) a->data int

c. (2 pts) &a Node **

d. (2 pts) a->next->next Node *

e. (2 pts) &x double *

f. (2 pts) d char *

g. (2 pts) *d char

h. (2 pts) argv[0] char *

i. (2 pts) argv[1][2] char

j. (2 pts) a->next Node *

k. (2 pts) x double

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt kiwi banana
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of argv[0][0]? .

c. (2 pts) What is the value of argv[1][2]? w

d. (2 pts) What is the value of argv[2][5]? a

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 1001 0100 from binary to base 10 148

   b. (2 pts) Convert 124 from base 10 to binary 0111 1100

   c. (2 pts) Convert 100 010 010 from base 2 to octal 422

   d. (2 pts) Convert 001 111 010 from binary to base 8 172

   e. (2 pts) Convert 0100 0001 1111 0010 from base 2 to base 16 41f2

   f. (2 pts) Convert 100 111 100 from binary to octal 474

   g. (2 pts) Convert 1010 0101 1101 0000 from base 2 to hexadecimal a5d0

   h. (2 pts) Convert 219 from base 10 to binary 1101 1011

   i. (2 pts) Convert 011 100 111 from base 2 to octal 347

   j. (2 pts) Convert f4f6 from base 16 to base 2 1111 0100 1111 0110

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int r;
  double s;
  Node t;
  char w;
  int *x;
  double *y;
  Node *z;
  char *a;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) y *double **

b. (2 pts) *y *double*

c. (2 pts) argv[1][2] *char*

d. (2 pts) argv[0] *char **

e. (2 pts) z->next *Node **

f. (2 pts) &r *int **

g. (2 pts) z->data *int*

h. (2 pts) z->next->next *Node **

i. (2 pts) argc *int*

j. (2 pts) w *char*

k. (2 pts) &z *Node ***

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt lemon kiwi lime cherry
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of argv[1][4]? n

c. (2 pts) What is the value of argv[0][5]? I

d. (2 pts) What is the value of argv[2][0]? k

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 110 001 011 from base 2 to octal 613

   b. (2 pts) Convert 4e37 from hexadecimal to base 2 0100 1110 0011 0111

   c. (2 pts) Convert 1011 1111 0110 1001 from base 2 to hexadecimal bf69

   d. (2 pts) Convert 1001 1001 from binary to decimal 153

   e. (2 pts) Convert 9810 from base 16 to base 2 1001 1000 0001 0000

   f. (2 pts) Convert d0da from hexadecimal to base 2 1101 0000 1101 1010

   g. (2 pts) Convert a7fe from base 16 to binary 1010 0111 1111 1110

   h. (2 pts) Convert 68 from decimal to base 2 0100 0100

   i. (2 pts) Convert 5 from octal to base 2 101

   j. (2 pts) Convert 2137 from base 16 to base 2 0010 0001 0011 0111

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  Node c;
  int d;
  double e;
  char f;
  Node *g;
  int *h;
  double *p;
  char *q;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) g->next Node *

b. (2 pts) argv[0] char *

c. (2 pts) &e double *

d. (2 pts) &g Node **

e. (2 pts) g->next->next Node *

f. (2 pts) c Node

g. (2 pts) g->data int

h. (2 pts) argc int

i. (2 pts) h int *

j. (2 pts) argv[1][2] char

k. (2 pts) *g Node

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt fig banana apple
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of argv[0][5]? I

c. (2 pts) What is the value of argv[2][5]? a

d. (2 pts) What is the value of argv[1][2]? g

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 10 from decimal to binary 1010

   b. (2 pts) Convert 1000 0110 0000 from binary to hexadecimal 860

   c. (2 pts) Convert 173 from decimal to binary 1010 1101

   d. (2 pts) Convert 37 from decimal to base 2 0010 0101

   e. (2 pts) Convert 36 from base 8 to binary 011 110

   f. (2 pts) Convert 106 from base 10 to base 2 0110 1010

   g. (2 pts) Convert 51ef from hexadecimal to binary 0101 0001 1110 1111

   h. (2 pts) Convert 62 from base 8 to base 2 110 010

   i. (2 pts) Convert 110 111 111 from base 2 to octal 677

   j. (2 pts) Convert bd21 from base 16 to base 2 1011 1101 0010 0001

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int z;
  Node a;
  double b;
  char c;
  int *d;
  Node *e;
  double *f;
  char *g;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) argc *int*

b. (2 pts) e->data *int*

c. (2 pts) argv[0] *char **

d. (2 pts) &f *double***

e. (2 pts) *f *double*

f. (2 pts) d *int **

g. (2 pts) b *double*

h. (2 pts) &b *double **

i. (2 pts) e->next->next *Node **

j. (2 pts) argv[1][2] *char*

k. (2 pts) e->next *Node **

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt kiwi grape banana
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of argv[1][2]? w

c. (2 pts) What is the value of argv[2][4]? e

d. (2 pts) What is the value of argv[0][1]? /

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 62e0 from base 16 to binary 0110 0010 1110 0000

   b. (2 pts) Convert ab2c from base 16 to base 2 1010 1011 0010 1100

   c. (2 pts) Convert 52ab from base 16 to base 2 0101 0010 1010 1011

   d. (2 pts) Convert 70 from base 8 to binary 111 000

   e. (2 pts) Convert ebd4 from base 16 to base 2 1110 1011 1101 0100

   f. (2 pts) Convert 011 011 000 from base 2 to base 8 330

   g. (2 pts) Convert 101 000 111 from binary to base 8 507

   h. (2 pts) Convert 114 from base 10 to binary 0111 0010

   i. (2 pts) Convert 1101 0010 0100 0001 from binary to base 16 d241

   j. (2 pts) Convert 200 from base 10 to binary 1100 1000

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double q;
  Node r;
  int s;
  char t;
  double *w;
  Node *x;
  int *y;
  char *z;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) argc *int*

b. (2 pts) x->next->next *Node **

c. (2 pts) s *int*

d. (2 pts) argv[1][2] *char*

e. (2 pts) x->data *int*

f. (2 pts) argv[0] *char **

g. (2 pts) *x *Node*

h. (2 pts) x->next *Node **

i. (2 pts) y *int **

j. (2 pts) &x *Node ***

k. (2 pts) &r *Node **

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt kiwi apple
```

a. (2 pts) What is the value of `argc` in this case? *3*

b. (2 pts) What is the value of argv[2][4]? *e*

c. (2 pts) What is the value of argv[1][3]? *i*

d. (2 pts) What is the value of argv[0][0]? *.*

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 51 from octal to binary 101 001

   b. (2 pts) Convert 011 001 010 from base 2 to octal 312

   c. (2 pts) Convert 0100 0000 from binary to base 10 64

   d. (2 pts) Convert 0110 1100 1101 1101 from base 2 to hexadecimal 6cdd

   e. (2 pts) Convert 63 from base 8 to base 2 110 011

   f. (2 pts) Convert 1 from base 8 to binary 001

   g. (2 pts) Convert 010 011 011 from base 2 to octal 233

   h. (2 pts) Convert 250 from base 10 to binary 1111 1010

   i. (2 pts) Convert 47 from base 8 to binary 100 111

   j. (2 pts) Convert 100 from decimal to binary 0110 0100

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int g;
  double h;
  Node p;
  char q;
  int *r;
  double *s;
  Node *t;
  char *w;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) *w *char*

b. (2 pts) t->data *int*

c. (2 pts) t->next *Node **

d. (2 pts) s *double **

e. (2 pts) &w *char ***

f. (2 pts) t->next->next *Node **

g. (2 pts) argv[1][2] *char*

h. (2 pts) argv[0] *char **

i. (2 pts) g *int*

j. (2 pts) &g *int **

k. (2 pts) argc *int*

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt kiwi mango lime fig
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of argv[2][1]? a

c. (2 pts) What is the value of argv[0][3]? u

d. (2 pts) What is the value of argv[1][1]? i

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 1101 1001 0010 0001 from base 2 to base 16 d921

   b. (2 pts) Convert 15 from octal to base 2 001 101

   c. (2 pts) Convert 0111 0110 1110 1011 from binary to base 16 76eb

   d. (2 pts) Convert 1100 1001 from binary to decimal 201

   e. (2 pts) Convert 001 001 010 from base 2 to octal 112

   f. (2 pts) Convert 0011 1000 from binary to base 10 56

   g. (2 pts) Convert 010 011 111 from binary to base 8 237

   h. (2 pts) Convert 30 from octal to binary 011 000

   i. (2 pts) Convert 17 from octal to base 2 001 111

   j. (2 pts) Convert 144 from decimal to binary 1001 0000

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  Node h;
  int p;
  double q;
  char r;
  Node *s;
  int *t;
  double *w;
  char *x;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) &r *char **

b. (2 pts) argc *int*

c. (2 pts) s *Node **

d. (2 pts) s->next->next *Node **

e. (2 pts) argv[1][2] *char*

f. (2 pts) &s *Node ***

g. (2 pts) s->next *Node **

h. (2 pts) p *int*

i. (2 pts) *w *double*

j. (2 pts) s->data *int*

k. (2 pts) argv[0] *char **

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt cherry fig lemon
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of argv[0][3]? u

c. (2 pts) What is the value of argv[2][2]? g

d. (2 pts) What is the value of argv[1][2]? e

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 111 100 from base 2 to octal 74

   b. (2 pts) Convert 111 100 010 from binary to octal 742

   c. (2 pts) Convert 31 from base 8 to base 2 011 001

   d. (2 pts) Convert 25 from base 8 to base 2 010 101

   e. (2 pts) Convert 8 from base 10 to binary 1000

   f. (2 pts) Convert 110 100 100 from base 2 to base 8 644

   g. (2 pts) Convert 111 110 011 from base 2 to base 8 763

   h. (2 pts) Convert 72 from octal to binary 111 010

   i. (2 pts) Convert 2 from base 8 to binary 010

   j. (2 pts) Convert 44 from decimal to binary 0010 1100

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  Node t;
  int w;
  double x;
  char y;
  Node *z;
  int *a;
  double *b;
  char *c;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) z->next Node *

b. (2 pts) argv[1][2] char

c. (2 pts) argc int

d. (2 pts) z->data int

e. (2 pts) y char

f. (2 pts) &c char **

g. (2 pts) argv[0] char *

h. (2 pts) *z Node

i. (2 pts) &x double *

j. (2 pts) c char *

k. (2 pts) z->next->next Node *

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt guava cherry mango
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of argv[1][2]? a

c. (2 pts) What is the value of argv[2][1]? h

d. (2 pts) What is the value of argv[0][2]? r

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 011 101 100 from base 2 to octal 354

   b. (2 pts) Convert 1001 0011 1100 1111 from binary to base 16 93cf

   c. (2 pts) Convert 1010 0010 1101 from binary to hexadecimal a2d

   d. (2 pts) Convert 109c from base 16 to base 2 0001 0000 1001 1100

   e. (2 pts) Convert 0111 1001 0000 0010 from base 2 to base 16 7902

   f. (2 pts) Convert 211 from base 10 to base 2 1101 0011

   g. (2 pts) Convert 0100 1011 from binary to decimal 75

   h. (2 pts) Convert 145 from decimal to binary 1001 0001

   i. (2 pts) Convert 76 from octal to base 2 111 110

   j. (2 pts) Convert 0011 1000 from binary to base 10 56

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double g;
  Node h;
  int p;
  char q;
  double *r;
  Node *s;
  int *t;
  char *w;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) argv[1][2] *char*

b. (2 pts) q *char*

c. (2 pts) s->next->next *Node **

d. (2 pts) &p *int **

e. (2 pts) s->data *int*

f. (2 pts) s->next *Node **

g. (2 pts) t *int **

h. (2 pts) &s *Node ***

i. (2 pts) argv[0] *char **

j. (2 pts) argc *int*

k. (2 pts) *w *char*

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt grape banana
```

a. (2 pts) What is the value of `argc` in this case? *3*

b. (2 pts) What is the value of argv[1][1]? *r*

c. (2 pts) What is the value of argv[0][4]? *n*

d. (2 pts) What is the value of argv[2][4]? *n*

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert bb3c from base 16 to binary 1011 1011 0011 1100

   b. (2 pts) Convert 0100 1101 from base 2 to base 10 77

   c. (2 pts) Convert 1111 1000 0000 1101 from base 2 to hexadecimal f80d

   d. (2 pts) Convert 9c86 from base 16 to binary 1001 1100 1000 0110

   e. (2 pts) Convert 010 111 000 from base 2 to octal 270

   f. (2 pts) Convert 0110 1101 from base 2 to decimal 109

   g. (2 pts) Convert f555 from base 16 to base 2 1111 0101 0101 0101

   h. (2 pts) Convert 0001 1000 1101 0100 from binary to hexadecimal 18d4

   i. (2 pts) Convert 1100 0100 1011 0101 from base 2 to base 16 c4b5

   j. (2 pts) Convert 1101 0100 from base 2 to base 10 212

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double c;
  int d;
  Node e;
  char f;
  double *g;
  int *h;
  Node *p;
  char *q;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) &g double **

b. (2 pts) p->next->next Node *

c. (2 pts) argv[0] char *

d. (2 pts) &d int *

e. (2 pts) *h int

f. (2 pts) h int *

g. (2 pts) p->next Node *

h. (2 pts) d int

i. (2 pts) argc int

j. (2 pts) argv[1][2] char

k. (2 pts) p->data int

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt mango banana kiwi grape
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of argv[0][0]? .

c. (2 pts) What is the value of argv[1][4]? o

d. (2 pts) What is the value of argv[2][2]? n

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 1110 1100 from binary to base 10 236

   b. (2 pts) Convert 111 110 from binary to base 8 76

   c. (2 pts) Convert 0010 1110 from base 2 to base 10 46

   d. (2 pts) Convert 1111 1000 1100 1000 from base 2 to base 16 f8c8

   e. (2 pts) Convert 1011 0010 from base 2 to decimal 178

   f. (2 pts) Convert 1001 1111 1100 0001 from binary to hexadecimal 9fc1

   g. (2 pts) Convert 111 101 111 from binary to octal 757

   h. (2 pts) Convert 40 from base 8 to binary 100 000

   i. (2 pts) Convert 011 001 100 from binary to base 8 314

   j. (2 pts) Convert 0 from base 10 to binary 0000

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int d;
  Node e;
  double f;
  char g;
  int *h;
  Node *p;
  double *q;
  char *r;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) p->next->next *Node **

b. (2 pts) f *double*

c. (2 pts) p->data *int*

d. (2 pts) h *int **

e. (2 pts) argv[0] *char **

f. (2 pts) argc *int*

g. (2 pts) &p *Node ***

h. (2 pts) &g *char **

i. (2 pts) argv[1][2] *char*

j. (2 pts) *h *int*

k. (2 pts) p->next *Node **

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt lime grape banana
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of argv[0][0]? .

c. (2 pts) What is the value of argv[2][3]? p

d. (2 pts) What is the value of argv[1][1]? i

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 0011 0001 0111 1110 from binary to base 16 *317e*

   b. (2 pts) Convert d9a7 from hexadecimal to binary *1101 1001 1010 0111*

   c. (2 pts) Convert 7 from base 8 to binary *111*

   d. (2 pts) Convert 132 from base 10 to binary *1000 0100*

   e. (2 pts) Convert 149 from base 10 to base 2 *1001 0101*

   f. (2 pts) Convert 399e from hexadecimal to base 2 *0011 1001 1001 1110*

   g. (2 pts) Convert 1010 0001 from base 2 to decimal *161*

   h. (2 pts) Convert 1001 from binary to decimal *9*

   i. (2 pts) Convert 14 from base 8 to binary *001 100*

   j. (2 pts) Convert 156 from base 10 to base 2 *1001 1100*

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int a;
  Node b;
  double c;
  char d;
  int *e;
  Node *f;
  double *g;
  char *h;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) &h *char* **

b. (2 pts) argc *int*

c. (2 pts) *g *double*

d. (2 pts) &d *char* *

e. (2 pts) h *char* *

f. (2 pts) f->next->next *Node* *

g. (2 pts) f->data *int*

h. (2 pts) argv[1][2] *char*

i. (2 pts) f->next *Node* *

j. (2 pts) argv[0] *char* *

k. (2 pts) b *Node*

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt lemon mango kiwi
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of argv[2][4]? o

c. (2 pts) What is the value of argv[1][1]? e

d. (2 pts) What is the value of argv[0][6]? t

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 11 from base 8 to binary 001 001

   b. (2 pts) Convert 110 000 011 from binary to base 8 603

   c. (2 pts) Convert 7dba from hexadecimal to base 2 0111 1101 1011 1010

   d. (2 pts) Convert bd8b from hexadecimal to binary 1011 1101 1000 1011

   e. (2 pts) Convert 33 from base 8 to binary 011 011

   f. (2 pts) Convert 35 from octal to binary 011 101

   g. (2 pts) Convert 14 from base 8 to binary 001 100

   h. (2 pts) Convert 1010 0101 1111 0000 from binary to base 16 a5f0

   i. (2 pts) Convert 65de from base 16 to base 2 0110 0101 1101 1110

   j. (2 pts) Convert 2555 from hexadecimal to binary 0010 0101 0101 0101

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double f;
  Node g;
  int h;
  char p;
  double *q;
  Node *r;
  int *s;
  char *t;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) r->next *Node **

b. (2 pts) &q *double ***

c. (2 pts) argv[0] *char **

d. (2 pts) g *Node*

e. (2 pts) r->next->next *Node **

f. (2 pts) *q *double*

g. (2 pts) t *char **

h. (2 pts) &h *int **

i. (2 pts) argv[1][2] *char*

j. (2 pts) r->data *int*

k. (2 pts) argc *int*

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt lemon apple
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of argv[0][6]? t

c. (2 pts) What is the value of argv[1][4]? n

d. (2 pts) What is the value of argv[2][2]? p

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 0110 1010 from base 2 to decimal 106

   b. (2 pts) Convert 0111 1011 from base 2 to decimal 123

   c. (2 pts) Convert 0110 1011 from binary to base 10 107

   d. (2 pts) Convert 0100 1001 from binary to base 10 73

   e. (2 pts) Convert 010 100 101 from base 2 to octal 245

   f. (2 pts) Convert 0001 0000 0110 1000 from base 2 to base 16 1068

   g. (2 pts) Convert 110 110 110 from binary to octal 666

   h. (2 pts) Convert 0010 1101 1010 0001 from binary to base 16 2da1

   i. (2 pts) Convert 47 from decimal to binary 0010 1111

   j. (2 pts) Convert c140 from base 16 to binary 1100 0001 0100 0000

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double c;
  int d;
  Node e;
  char f;
  double *g;
  int *h;
  Node *p;
  char *q;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) &d *int **

b. (2 pts) argv[0] *char **

c. (2 pts) p->data *int*

d. (2 pts) f *char*

e. (2 pts) p->next->next *Node **

f. (2 pts) *p *Node*

g. (2 pts) &p *Node ***

h. (2 pts) argv[1][2] *char*

i. (2 pts) argc *int*

j. (2 pts) h *int **

k. (2 pts) p->next *Node **

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt lemon grape apple banana
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of argv[2][2]? a

c. (2 pts) What is the value of argv[0][5]? I

d. (2 pts) What is the value of argv[1][1]? e

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 100 111 000 from binary to base 8 470

   b. (2 pts) Convert 77 from base 10 to binary 0100 1101

   c. (2 pts) Convert 1010 0001 1111 1010 from binary to base 16 a1fa

   d. (2 pts) Convert 1010 0101 1011 0111 from binary to hexadecimal a5b7

   e. (2 pts) Convert 52 from base 8 to binary 101 010

   f. (2 pts) Convert 20 from octal to base 2 010 000

   g. (2 pts) Convert 67 from base 8 to binary 110 111

   h. (2 pts) Convert 45 from base 8 to base 2 100 101

   i. (2 pts) Convert d1b0 from hexadecimal to base 2 1101 0001 1011 0000

   j. (2 pts) Convert ed81 from base 16 to base 2 1110 1101 1000 0001

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int c;
  Node d;
  double e;
  char f;
  int *g;
  Node *h;
  double *p;
  char *q;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) *h Node

b. (2 pts) argc int

c. (2 pts) c int

d. (2 pts) &q char **

e. (2 pts) argv[0] char *

f. (2 pts) &c int *

g. (2 pts) argv[1][2] char

h. (2 pts) h->next->next Node *

i. (2 pts) h Node *

j. (2 pts) h->next Node *

k. (2 pts) h->data int

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt banana guava grape
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of argv[0][3]? u

c. (2 pts) What is the value of argv[2][1]? u

d. (2 pts) What is the value of argv[1][3]? a

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 224 from decimal to base 2 *1110 0000*

   b. (2 pts) Convert 763 from hexadecimal to base 2 *0111 0110 0011*

   c. (2 pts) Convert 8fda from base 16 to base 2 *1000 1111 1101 1010*

   d. (2 pts) Convert 49 from decimal to base 2 *0011 0001*

   e. (2 pts) Convert 1000 1100 from base 2 to base 10 *140*

   f. (2 pts) Convert 110 111 001 from binary to octal *671*

   g. (2 pts) Convert 41 from octal to binary *100 001*

   h. (2 pts) Convert 0001 1110 from binary to base 10 *30*

   i. (2 pts) Convert 155 from base 10 to binary *1001 1011*

   j. (2 pts) Convert 896b from base 16 to binary *1000 1001 0110 1011*

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  Node z;
  double a;
  int b;
  char c;
  Node *d;
  double *e;
  int *f;
  char *g;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) &c *char **

b. (2 pts) d->data *int*

c. (2 pts) *g *char*

d. (2 pts) argv[1][2] *char*

e. (2 pts) d->next->next *Node **

f. (2 pts) argv[0] *char **

g. (2 pts) &e *double ***

h. (2 pts) argc *int*

i. (2 pts) d->next *Node **

j. (2 pts) c *char*

k. (2 pts) d *Node **

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt date guava
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of argv[1][2]? t

c. (2 pts) What is the value of argv[2][1]? u

d. (2 pts) What is the value of argv[0][3]? u

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 57 from decimal to base 2 0011 1001

   b. (2 pts) Convert 1010 1010 from binary to base 10 170

   c. (2 pts) Convert 0011 0101 0011 1100 from binary to hexadecimal 353c

   d. (2 pts) Convert 111 011 010 from base 2 to base 8 732

   e. (2 pts) Convert 77 from base 8 to binary 111 111

   f. (2 pts) Convert 221 from decimal to binary 1101 1101

   g. (2 pts) Convert 110 110 001 from base 2 to base 8 661

   h. (2 pts) Convert 1100 0100 1011 0010 from binary to base 16 c4b2

   i. (2 pts) Convert 142 from base 10 to base 2 1000 1110

   j. (2 pts) Convert 148 from base 10 to base 2 1001 0100

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  Node q;
  double r;
  int s;
  char t;
  Node *w;
  double *x;
  int *y;
  char *z;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) w->data *int*

b. (2 pts) w->next->next *Node **

c. (2 pts) w->next *Node **

d. (2 pts) y *int **

e. (2 pts) &t *char **

f. (2 pts) argv[0] *char **

g. (2 pts) s *int*

h. (2 pts) argc *int*

i. (2 pts) &w *Node ***

j. (2 pts) *y *int*

k. (2 pts) argv[1][2] *char*

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt date lime
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of argv[0][3]? u

c. (2 pts) What is the value of argv[1][0]? d

d. (2 pts) What is the value of argv[2][1]? i

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 37 from octal to binary 011 111

   b. (2 pts) Convert 0110 0100 0101 0111 from binary to hexadecimal 6457

   c. (2 pts) Convert 0010 0011 0001 1100 from base 2 to hexadecimal 231c

   d. (2 pts) Convert 011 110 010 from base 2 to octal 362

   e. (2 pts) Convert 1110 0000 from base 2 to base 10 224

   f. (2 pts) Convert 0111 0111 from base 2 to decimal 119

   g. (2 pts) Convert 130 from decimal to binary 1000 0010

   h. (2 pts) Convert 23 from octal to binary 010 011

   i. (2 pts) Convert 0101 1000 from binary to decimal 88

   j. (2 pts) Convert 48 from decimal to base 2 0011 0000

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int y;
  double z;
  Node a;
  char b;
  int *c;
  double *d;
  Node *e;
  char *f;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) argv[0] *char **

b. (2 pts) *c *int*

c. (2 pts) &e *Node ***

d. (2 pts) e->data *int*

e. (2 pts) y *int*

f. (2 pts) argv[1][2] *char*

g. (2 pts) e->next *Node **

h. (2 pts) d *double **

i. (2 pts) &a *Node **

j. (2 pts) argc *int*

k. (2 pts) e->next->next *Node **

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt grape fig mango lime
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of argv[1][4]? e

c. (2 pts) What is the value of argv[0][3]? u

d. (2 pts) What is the value of argv[2][1]? i

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert af82 from base 16 to base 2 1010 1111 1000 0010

   b. (2 pts) Convert 0011 0101 1101 1101 from base 2 to base 16 35dd

   c. (2 pts) Convert 26 from base 8 to base 2 010 110

   d. (2 pts) Convert 1101 0101 from binary to decimal 213

   e. (2 pts) Convert 54 from base 10 to base 2 0011 0110

   f. (2 pts) Convert 1010 1010 0100 0101 from base 2 to base 16 aa45

   g. (2 pts) Convert 100 001 010 from base 2 to base 8 412

   h. (2 pts) Convert 1011 0101 from base 2 to base 10 181

   i. (2 pts) Convert 250 from base 10 to base 2 1111 1010

   j. (2 pts) Convert 93 from decimal to binary 0101 1101

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  Node y;
  int z;
  double a;
  char b;
  Node *c;
  int *d;
  double *e;
  char *f;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) c->next Node *

b. (2 pts) argv[1][2] char

c. (2 pts) c Node *

d. (2 pts) argv[0] char *

e. (2 pts) c->data int

f. (2 pts) argc int

g. (2 pts) c->next->next Node *

h. (2 pts) &c Node **

i. (2 pts) *f char

j. (2 pts) &z int *

k. (2 pts) a double

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt cherry guava lemon
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of argv[0][1]? /

c. (2 pts) What is the value of argv[2][2]? a

d. (2 pts) What is the value of argv[1][0]? c

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 111 101 000 from binary to base 8 750

   b. (2 pts) Convert 111 100 000 from base 2 to octal 740

   c. (2 pts) Convert 21 from octal to binary 010 001

   d. (2 pts) Convert 97 from base 10 to base 2 0110 0001

   e. (2 pts) Convert 0001 1001 0111 1101 from base 2 to hexadecimal 197d

   f. (2 pts) Convert 4422 from hexadecimal to base 2 0100 0100 0010 0010

   g. (2 pts) Convert 001 011 110 from binary to base 8 136

   h. (2 pts) Convert 0011 1101 from base 2 to base 10 61

   i. (2 pts) Convert 1100 0100 from binary to decimal 196

   j. (2 pts) Convert 249 from decimal to binary 1111 1001

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  Node t;
  double w;
  int x;
  char y;
  Node *z;
  double *a;
  int *b;
  char *c;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) argv[0] *char **

b. (2 pts) z->next *Node **

c. (2 pts) *a *double*

d. (2 pts) c *char **

e. (2 pts) z->data *int*

f. (2 pts) argv[1][2] *char*

g. (2 pts) z->next->next *Node **

h. (2 pts) t *Node*

i. (2 pts) argc *int*

j. (2 pts) &t *Node **

k. (2 pts) &b *int ***

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt fig guava
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of argv[1][0]? f

c. (2 pts) What is the value of argv[2][2]? a

d. (2 pts) What is the value of argv[0][4]? n

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 010 011 001 from base 2 to base 8 231

   b. (2 pts) Convert 92d2 from hexadecimal to base 2 1001 0010 1101 0010

   c. (2 pts) Convert 111 011 001 from base 2 to base 8 731

   d. (2 pts) Convert 1cdd from base 16 to base 2 0001 1100 1101 1101

   e. (2 pts) Convert 100 010 011 from base 2 to base 8 423

   f. (2 pts) Convert 010 001 010 from binary to base 8 212

   g. (2 pts) Convert 80c0 from base 16 to binary 1000 0000 1100 0000

   h. (2 pts) Convert 1110 0011 0111 0011 from base 2 to hexadecimal e373

   i. (2 pts) Convert 1011 0110 from base 2 to decimal 182

   j. (2 pts) Convert 0100 from binary to base 10 4

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double f;
  Node g;
  int h;
  char p;
  double *q;
  Node *r;
  int *s;
  char *t;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) s *int **

b. (2 pts) p *char*

c. (2 pts) argv[0] *char **

d. (2 pts) &p *char **

e. (2 pts) argc *int*

f. (2 pts) &r *Node ***

g. (2 pts) *q *double*

h. (2 pts) r->data *int*

i. (2 pts) argv[1][2] *char*

j. (2 pts) r->next *Node **

k. (2 pts) r->next->next *Node **

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt mango lemon
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of argv[1][4]? o

c. (2 pts) What is the value of argv[0][4]? n

d. (2 pts) What is the value of argv[2][4]? n

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 919d from hexadecimal to base 2 1001 0001 1001 1101

   b. (2 pts) Convert 010 011 001 from binary to octal 231

   c. (2 pts) Convert 66 from base 8 to base 2 110 110

   d. (2 pts) Convert 1010 1000 1100 0111 from base 2 to hexadecimal a8c7

   e. (2 pts) Convert 109 from base 10 to base 2 0110 1101

   f. (2 pts) Convert 110 111 110 from binary to octal 676

   g. (2 pts) Convert 2ab0 from hexadecimal to base 2 0010 1010 1011 0000

   h. (2 pts) Convert 0110 1011 from binary to decimal 107

   i. (2 pts) Convert 128 from decimal to base 2 1000 0000

   j. (2 pts) Convert 1010 0000 from base 2 to decimal 160

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int c;
  double d;
  Node e;
  char f;
  int *g;
  double *h;
  Node *p;
  char *q;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) argv[1][2] *char*

b. (2 pts) p->data *int*

c. (2 pts) p->next *Node **

d. (2 pts) argv[0] *char **

e. (2 pts) h *double **

f. (2 pts) p->next->next *Node **

g. (2 pts) &e *Node **

h. (2 pts) *p *Node*

i. (2 pts) argc *int*

j. (2 pts) e *Node*

k. (2 pts) &q *char ***

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt banana apple grape guava
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of argv[0][5]? I

c. (2 pts) What is the value of argv[2][1]? p

d. (2 pts) What is the value of argv[1][0]? b

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 1100 0010 from base 2 to decimal 194

   b. (2 pts) Convert 7 from base 8 to base 2 111

   c. (2 pts) Convert 16 from base 10 to binary 0001 0000

   d. (2 pts) Convert 1 from base 8 to base 2 001

   e. (2 pts) Convert c35f from base 16 to base 2 1100 0011 0101 1111

   f. (2 pts) Convert 100 011 from base 2 to base 8 43

   g. (2 pts) Convert 44 from base 10 to binary 0010 1100

   h. (2 pts) Convert 1101 0100 from binary to decimal 212

   i. (2 pts) Convert 34 from base 10 to base 2 0010 0010

   j. (2 pts) Convert 204 from base 10 to base 2 1100 1100

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  Node t;
  int w;
  double x;
  char y;
  Node *z;
  int *a;
  double *b;
  char *c;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) z->data int

b. (2 pts) *a int

c. (2 pts) &a int **

d. (2 pts) z Node *

e. (2 pts) z->next Node *

f. (2 pts) z->next->next Node *

g. (2 pts) argc int

h. (2 pts) argv[0] char *

i. (2 pts) y char

j. (2 pts) argv[1][2] char

k. (2 pts) &w int *

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt date grape fig
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of argv[1][1]? a

c. (2 pts) What is the value of argv[2][0]? g

d. (2 pts) What is the value of argv[0][3]? u

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 0111 1101 1111 from base 2 to base 16 7df

   b. (2 pts) Convert 216 from base 10 to binary 1101 1000

   c. (2 pts) Convert 111 111 101 from binary to octal 775

   d. (2 pts) Convert 44 from base 8 to binary 100 100

   e. (2 pts) Convert 1010 0110 1010 1010 from base 2 to hexadecimal a6aa

   f. (2 pts) Convert 1010 1011 from binary to base 10 171

   g. (2 pts) Convert 214 from decimal to binary 1101 0110

   h. (2 pts) Convert 0101 1011 1100 0011 from base 2 to hexadecimal 5bc3

   i. (2 pts) Convert 236 from decimal to base 2 1110 1100

   j. (2 pts) Convert 104 from base 10 to base 2 0110 1000

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  Node q;
  int r;
  double s;
  char t;
  Node *w;
  int *x;
  double *y;
  char *z;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) z *char **

b. (2 pts) r *int*

c. (2 pts) *z *char*

d. (2 pts) argc *int*

e. (2 pts) &z *char ***

f. (2 pts) w->data *int*

g. (2 pts) w->next->next *Node **

h. (2 pts) &q *Node **

i. (2 pts) w->next *Node **

j. (2 pts) argv[0] *char **

k. (2 pts) argv[1][2] *char*

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt lemon lime fig
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of argv[2][1]? i

c. (2 pts) What is the value of argv[1][3]? o

d. (2 pts) What is the value of argv[0][1]? /

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 0110 0000 0011 1011 from base 2 to base 16 603b

   b. (2 pts) Convert 011 110 110 from base 2 to octal 366

   c. (2 pts) Convert 1010 0100 from base 2 to base 10 164

   d. (2 pts) Convert 4c85 from base 16 to binary 0100 1100 1000 0101

   e. (2 pts) Convert 0001 0111 from base 2 to base 10 23

   f. (2 pts) Convert ace6 from hexadecimal to base 2 1010 1100 1110 0110

   g. (2 pts) Convert 12 from base 8 to base 2 001 010

   h. (2 pts) Convert 0010 0011 0101 from base 2 to hexadecimal 235

   i. (2 pts) Convert 1101 1110 from base 2 to decimal 222

   j. (2 pts) Convert 0111 0100 from base 2 to decimal 116

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double c;
  Node d;
  int e;
  char f;
  double *g;
  Node *h;
  int *p;
  char *q;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) h->next Node *

b. (2 pts) argv[1][2] char

c. (2 pts) c double

d. (2 pts) h->data int

e. (2 pts) argv[0] char *

f. (2 pts) argc int

g. (2 pts) *p int

h. (2 pts) &c double *

i. (2 pts) h->next->next Node *

j. (2 pts) &p int **

k. (2 pts) h Node *

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt banana lemon
```

a. (2 pts) What is the value of `argc` in this case? *3*

b. (2 pts) What is the value of argv[1][5]? *a*

c. (2 pts) What is the value of argv[0][2]? *r*

d. (2 pts) What is the value of argv[2][2]? *m*

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 101 001 010 from base 2 to base 8 *512*

   b. (2 pts) Convert 0011 0101 from binary to base 10 *53*

   c. (2 pts) Convert 1001 0010 from binary to decimal *146*

   d. (2 pts) Convert 1101 1000 from base 2 to base 10 *216*

   e. (2 pts) Convert 250 from base 10 to binary *1111 1010*

   f. (2 pts) Convert 70 from decimal to binary *0100 0110*

   g. (2 pts) Convert 64 from octal to binary *110 100*

   h. (2 pts) Convert 1000 1001 from base 2 to decimal *137*

   i. (2 pts) Convert 1010 1001 from binary to base 10 *169*

   j. (2 pts) Convert 100 000 from binary to octal *40*

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double y;
  int z;
  Node a;
  char b;
  double *c;
  int *d;
  Node *e;
  char *f;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) argv[1][2] *char*

b. (2 pts) e->next->next *Node **

c. (2 pts) e->next *Node **

d. (2 pts) &c *double ***

e. (2 pts) *c *double*

f. (2 pts) b *char*

g. (2 pts) c *double **

h. (2 pts) argv[0] *char **

i. (2 pts) e->data *int*

j. (2 pts) &b *char **

k. (2 pts) argc *int*

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt grape guava apple cherry
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of argv[0][3]? u

c. (2 pts) What is the value of argv[2][4]? a

d. (2 pts) What is the value of argv[1][1]? r

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 65 from octal to binary 110 101

   b. (2 pts) Convert 723 from hexadecimal to base 2 0111 0010 0011

   c. (2 pts) Convert c87e from hexadecimal to base 2 1100 1000 0111 1110

   d. (2 pts) Convert 0011 0100 1011 0001 from base 2 to base 16 34b1

   e. (2 pts) Convert 508c from base 16 to binary 0101 0000 1000 1100

   f. (2 pts) Convert 36 from base 8 to binary 011 110

   g. (2 pts) Convert d4a1 from base 16 to base 2 1101 0100 1010 0001

   h. (2 pts) Convert 1111 0010 from binary to decimal 242

   i. (2 pts) Convert 0100 1010 from base 2 to decimal 74

   j. (2 pts) Convert 0011 1100 from base 2 to decimal 60

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int p;
  Node q;
  double r;
  char s;
  int *t;
  Node *w;
  double *x;
  char *y;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) argc int

b. (2 pts) argv[1][2] char

c. (2 pts) &w Node **

d. (2 pts) &r double *

e. (2 pts) t int *

f. (2 pts) w->next->next Node *

g. (2 pts) w->next Node *

h. (2 pts) p int

i. (2 pts) *y char

j. (2 pts) w->data int

k. (2 pts) argv[0] char *

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt guava lime fig
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of argv[2][0]? l

c. (2 pts) What is the value of argv[0][3]? u

d. (2 pts) What is the value of argv[1][1]? u

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 0001 1011 from binary to decimal 27

   b. (2 pts) Convert c14c from base 16 to binary 1100 0001 0100 1100

   c. (2 pts) Convert b65f from base 16 to base 2 1011 0110 0101 1111

   d. (2 pts) Convert 192 from base 10 to base 2 1100 0000

   e. (2 pts) Convert 14 from base 8 to base 2 001 100

   f. (2 pts) Convert 0001 0011 0000 1001 from base 2 to base 16 1309

   g. (2 pts) Convert 7e92 from hexadecimal to base 2 0111 1110 1001 0010

   h. (2 pts) Convert 0111 1010 1000 0100 from binary to hexadecimal 7a84

   i. (2 pts) Convert 20 from decimal to base 2 0001 0100

   j. (2 pts) Convert 1101 1000 from binary to decimal 216

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int f;
  Node g;
  double h;
  char p;
  int *q;
  Node *r;
  double *s;
  char *t;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) *r Node

b. (2 pts) r->next Node *

c. (2 pts) h double

d. (2 pts) &t char **

e. (2 pts) argv[0] char *

f. (2 pts) argv[1][2] char

g. (2 pts) r->next->next Node *

h. (2 pts) argc int

i. (2 pts) &g Node *

j. (2 pts) t char *

k. (2 pts) r->data int

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt lime apple banana
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of argv[2][4]? e

c. (2 pts) What is the value of argv[1][0]? l

d. (2 pts) What is the value of argv[0][4]? n

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 1101 0111 from base 2 to base 10 215

   b. (2 pts) Convert 0001 1110 1111 1111 from base 2 to base 16 1eff

   c. (2 pts) Convert 159 from base 10 to binary 1001 1111

   d. (2 pts) Convert 77 from base 8 to binary 111 111

   e. (2 pts) Convert 0011 1010 1110 0101 from base 2 to hexadecimal 3ae5

   f. (2 pts) Convert d8bd from base 16 to base 2 1101 1000 1011 1101

   g. (2 pts) Convert 9231 from hexadecimal to binary 1001 0010 0011 0001

   h. (2 pts) Convert 12 from octal to base 2 001 010

   i. (2 pts) Convert 1100 0100 0110 1100 from binary to hexadecimal c46c

   j. (2 pts) Convert 102 from base 10 to base 2 0110 0110

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double d;
  int e;
  Node f;
  char g;
  double *h;
  int *p;
  Node *q;
  char *r;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) &d *double **

b. (2 pts) argv[1][2] *char*

c. (2 pts) h *double **

d. (2 pts) *p *int*

e. (2 pts) q->next->next *Node **

f. (2 pts) &p *int ***

g. (2 pts) q->data *int*

h. (2 pts) q->next *Node **

i. (2 pts) d *double*

j. (2 pts) argc *int*

k. (2 pts) argv[0] *char **

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt grape banana date mango
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of argv[2][3]? a

c. (2 pts) What is the value of argv[1][3]? p

d. (2 pts) What is the value of argv[0][3]? u

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 0001 1100 1010 1110 from binary to hexadecimal 1cae

   b. (2 pts) Convert 66 from octal to binary 110 110

   c. (2 pts) Convert 100 011 011 from binary to octal 433

   d. (2 pts) Convert 1000 1010 1101 0010 from base 2 to base 16 8ad2

   e. (2 pts) Convert 0001 1110 from base 2 to decimal 30

   f. (2 pts) Convert 114 from decimal to binary 0111 0010

   g. (2 pts) Convert 17 from base 8 to binary 001 111

   h. (2 pts) Convert 1011 0010 from binary to base 10 178

   i. (2 pts) Convert 8e83 from base 16 to base 2 1000 1110 1000 0011

   j. (2 pts) Convert 2 from base 10 to base 2 0010

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int q;
  double r;
  Node s;
  char t;
  int *w;
  double *x;
  Node *y;
  char *z;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) z *char **

b. (2 pts) argv[1][2] *char*

c. (2 pts) &t *char **

d. (2 pts) argv[0] *char **

e. (2 pts) &z *char ***

f. (2 pts) *y *Node*

g. (2 pts) t *char*

h. (2 pts) y->next->next *Node **

i. (2 pts) y->next *Node **

j. (2 pts) argc *int*

k. (2 pts) y->data *int*

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt date grape apple kiwi
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of argv[0][4]? n

c. (2 pts) What is the value of argv[2][3]? p

d. (2 pts) What is the value of argv[1][0]? d

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 78 from base 10 to base 2 *0100 1110*

   b. (2 pts) Convert 101 010 101 from binary to base 8 *525*

   c. (2 pts) Convert c3f4 from hexadecimal to base 2 *1100 0011 1111 0100*

   d. (2 pts) Convert 71 from base 8 to binary *111 001*

   e. (2 pts) Convert 011 101 000 from base 2 to base 8 *350*

   f. (2 pts) Convert a503 from base 16 to base 2 *1010 0101 0000 0011*

   g. (2 pts) Convert 3e50 from hexadecimal to base 2 *0011 1110 0101 0000*

   h. (2 pts) Convert 27 from base 10 to base 2 *0001 1011*

   i. (2 pts) Convert 0011 0000 0011 1110 from base 2 to hexadecimal *303e*

   j. (2 pts) Convert 46 from base 10 to binary *0010 1110*

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  Node q;
  int r;
  double s;
  char t;
  Node *w;
  int *x;
  double *y;
  char *z;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) argv[0] *char **

b. (2 pts) *y *double*

c. (2 pts) w->next *Node **

d. (2 pts) &w *Node ***

e. (2 pts) w->next->next *Node **

f. (2 pts) w->data *int*

g. (2 pts) argc *int*

h. (2 pts) argv[1][2] *char*

i. (2 pts) y *double **

j. (2 pts) &s *double **

k. (2 pts) q *Node*

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt banana kiwi lemon
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of argv[2][1]? i

c. (2 pts) What is the value of argv[1][1]? a

d. (2 pts) What is the value of argv[0][2]? r

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 44 from base 8 to base 2 *100 100*

   b. (2 pts) Convert 0110 0100 from base 2 to decimal *100*

   c. (2 pts) Convert 1011 0001 from binary to decimal *177*

   d. (2 pts) Convert 34 from base 8 to base 2 *011 100*

   e. (2 pts) Convert 87 from base 10 to base 2 *0101 0111*

   f. (2 pts) Convert 0011 1110 1110 0000 from base 2 to base 16 *3ee0*

   g. (2 pts) Convert e840 from base 16 to binary *1110 1000 0100 0000*

   h. (2 pts) Convert 1010 0011 0011 1010 from base 2 to base 16 *a33a*

   i. (2 pts) Convert fa55 from hexadecimal to base 2 *1111 1010 0101 0101*

   j. (2 pts) Convert 202 from decimal to binary *1100 1010*

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double g;
  Node h;
  int p;
  char q;
  double *r;
  Node *s;
  int *t;
  char *w;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) s->next Node *

b. (2 pts) s->data int

c. (2 pts) argv[0] char *

d. (2 pts) argc int

e. (2 pts) p int

f. (2 pts) argv[1][2] char

g. (2 pts) s Node *

h. (2 pts) s->next->next Node *

i. (2 pts) &h Node *

j. (2 pts) *w char

k. (2 pts) &w char **

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt apple guava
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of argv[1][0]? a

c. (2 pts) What is the value of argv[2][0]? g

d. (2 pts) What is the value of argv[0][6]? t

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 72 from base 8 to base 2 111 010

   b. (2 pts) Convert 1 from base 8 to base 2 001

   c. (2 pts) Convert 87 from decimal to base 2 0101 0111

   d. (2 pts) Convert 2e91 from base 16 to binary 0010 1110 1001 0001

   e. (2 pts) Convert 110 010 000 from base 2 to octal 620

   f. (2 pts) Convert 010 000 000 from base 2 to base 8 200

   g. (2 pts) Convert 001 110 011 from base 2 to base 8 163

   h. (2 pts) Convert 0100 1001 from base 2 to base 10 73

   i. (2 pts) Convert ecc8 from hexadecimal to base 2 1110 1100 1100 1000

   j. (2 pts) Convert 1101 0110 from base 2 to base 10 214

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double z;
  int a;
  Node b;
  char c;
  double *d;
  int *e;
  Node *f;
  char *g;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) d *double **

b. (2 pts) argv[1][2] *char*

c. (2 pts) argv[0] *char **

d. (2 pts) argc *int*

e. (2 pts) f->data *int*

f. (2 pts) b *Node*

g. (2 pts) &a *int **

h. (2 pts) &e *int ***

i. (2 pts) *g *char*

j. (2 pts) f->next->next *Node **

k. (2 pts) f->next *Node **

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt cherry kiwi mango fig
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of argv[1][0]? c

c. (2 pts) What is the value of argv[0][4]? n

d. (2 pts) What is the value of argv[2][0]? k

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 48 from base 10 to base 2 *0011 0000*

   b. (2 pts) Convert 193 from base 10 to base 2 *1100 0001*

   c. (2 pts) Convert 4516 from hexadecimal to base 2 *0100 0101 0001 0110*

   d. (2 pts) Convert 1011 1010 from base 2 to decimal *186*

   e. (2 pts) Convert 171 from base 10 to base 2 *1010 1011*

   f. (2 pts) Convert 66 from octal to binary *110 110*

   g. (2 pts) Convert 111 000 111 from binary to base 8 *707*

   h. (2 pts) Convert 209 from decimal to base 2 *1101 0001*

   i. (2 pts) Convert 1011 0110 1110 0000 from base 2 to hexadecimal *b6e0*

   j. (2 pts) Convert 0111 0010 from binary to decimal *114*

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double w;
  int x;
  Node y;
  char z;
  double *a;
  int *b;
  Node *c;
  char *d;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) &a double **

b. (2 pts) argv[0] char *

c. (2 pts) c->next Node *

d. (2 pts) argc int

e. (2 pts) *a double

f. (2 pts) d char *

g. (2 pts) c->next->next Node *

h. (2 pts) argv[1][2] char

i. (2 pts) &w double *

j. (2 pts) c->data int

k. (2 pts) w double

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt date lime fig mango
```

a. (2 pts) What is the value of `argc` in this case? *5*

b. (2 pts) What is the value of argv[0][2]? *r*

c. (2 pts) What is the value of argv[2][2]? *m*

d. (2 pts) What is the value of argv[1][1]? *a*

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 011 000 011 from binary to octal 303

   b. (2 pts) Convert 1001 0011 from binary to base 10 147

   c. (2 pts) Convert 011 110 110 from binary to octal 366

   d. (2 pts) Convert 0001 0110 1011 1100 from base 2 to hexadecimal 16bc

   e. (2 pts) Convert 17b from base 16 to base 2 0001 0111 1011

   f. (2 pts) Convert 1100 from binary to decimal 12

   g. (2 pts) Convert 111 001 100 from base 2 to octal 714

   h. (2 pts) Convert 0011 1010 0100 1011 from base 2 to base 16 3a4b

   i. (2 pts) Convert 589a from hexadecimal to base 2 0101 1000 1001 1010

   j. (2 pts) Convert 1001 1110 from binary to base 10 158

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int f;
  Node g;
  double h;
  char p;
  int *q;
  Node *r;
  double *s;
  char *t;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) *q int

b. (2 pts) argv[0] char *

c. (2 pts) r->next->next Node *

d. (2 pts) g Node

e. (2 pts) r->next Node *

f. (2 pts) argc int

g. (2 pts) &r Node **

h. (2 pts) s double *

i. (2 pts) r->data int

j. (2 pts) argv[1][2] char

k. (2 pts) &p char *

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt grape date cherry
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of argv[2][0]? d

c. (2 pts) What is the value of argv[0][5]? I

d. (2 pts) What is the value of argv[1][4]? e

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert a66d from hexadecimal to base 2 *1010 0110 0110 1101*

   b. (2 pts) Convert 77 from decimal to binary *0100 1101*

   c. (2 pts) Convert 0110 1001 0101 0110 from base 2 to hexadecimal *6956*

   d. (2 pts) Convert 162 from base 10 to base 2 *1010 0010*

   e. (2 pts) Convert 71 from base 8 to binary *111 001*

   f. (2 pts) Convert 101 001 100 from base 2 to base 8 *514*

   g. (2 pts) Convert 100 100 000 from binary to base 8 *440*

   h. (2 pts) Convert 60 from base 8 to base 2 *110 000*

   i. (2 pts) Convert 34 from base 10 to binary *0010 0010*

   j. (2 pts) Convert 0011 1010 from base 2 to base 10 *58*

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  Node c;
  double d;
  int e;
  char f;
  Node *g;
  double *h;
  int *p;
  char *q;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) &e *int **

b. (2 pts) &q *char ***

c. (2 pts) *h *double*

d. (2 pts) argv[1][2] *char*

e. (2 pts) h *double **

f. (2 pts) g->data *int*

g. (2 pts) f *char*

h. (2 pts) argv[0] *char **

i. (2 pts) g->next->next *Node **

j. (2 pts) g->next *Node **

k. (2 pts) argc *int*

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt guava lime
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of argv[2][0]? l

c. (2 pts) What is the value of argv[1][1]? u

d. (2 pts) What is the value of argv[0][0]? .

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert fec9 from hexadecimal to base 2 1111 1110 1100 1001

   b. (2 pts) Convert 1111 0000 from base 2 to decimal 240

   c. (2 pts) Convert 3 from octal to base 2 011

   d. (2 pts) Convert 010 111 100 from binary to base 8 274

   e. (2 pts) Convert 0101 0101 from base 2 to base 10 85

   f. (2 pts) Convert 1010 0111 1010 0100 from base 2 to base 16 a7a4

   g. (2 pts) Convert 225 from base 10 to binary 1110 0001

   h. (2 pts) Convert 0110 1000 from base 2 to base 10 104

   i. (2 pts) Convert 21 from decimal to base 2 0001 0101

   j. (2 pts) Convert 010 001 100 from binary to octal 214

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int t;
  double w;
  Node x;
  char y;
  int *z;
  double *a;
  Node *b;
  char *c;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) c *char **

b. (2 pts) &b *Node ***

c. (2 pts) y *char*

d. (2 pts) *a *double*

e. (2 pts) argv[0] *char **

f. (2 pts) b->next *Node **

g. (2 pts) b->data *int*

h. (2 pts) argc *int*

i. (2 pts) argv[1][2] *char*

j. (2 pts) &w *double **

k. (2 pts) b->next->next *Node **

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt banana grape guava mango
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of argv[1][5]? a

c. (2 pts) What is the value of argv[0][0]? .

d. (2 pts) What is the value of argv[2][4]? e

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 20 from octal to base 2 010 000

   b. (2 pts) Convert 1010 1010 0110 1110 from base 2 to base 16 aa6e

   c. (2 pts) Convert fc98 from base 16 to binary 1111 1100 1001 1000

   d. (2 pts) Convert 111 010 100 from binary to octal 724

   e. (2 pts) Convert 388b from base 16 to base 2 0011 1000 1000 1011

   f. (2 pts) Convert 4181 from base 16 to base 2 0100 0001 1000 0001

   g. (2 pts) Convert 8ba6 from hexadecimal to binary 1000 1011 1010 0110

   h. (2 pts) Convert 1111 0000 0001 1110 from base 2 to base 16 f01e

   i. (2 pts) Convert df3c from base 16 to binary 1101 1111 0011 1100

   j. (2 pts) Convert 111 000 011 from base 2 to octal 703

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int q;
  double r;
  Node s;
  char t;
  int *w;
  double *x;
  Node *y;
  char *z;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) y *Node **

b. (2 pts) &w *int ***

c. (2 pts) y->data *int*

d. (2 pts) argv[1][2] *char*

e. (2 pts) r *double*

f. (2 pts) argc *int*

g. (2 pts) argv[0] *char **

h. (2 pts) &q *int **

i. (2 pts) y->next *Node **

j. (2 pts) y->next->next *Node **

k. (2 pts) *y *Node*

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt apple kiwi cherry date
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of argv[0][1]? /

c. (2 pts) What is the value of argv[1][0]? a

d. (2 pts) What is the value of argv[2][2]? w

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 0111 0101 0000 1011 from base 2 to hexadecimal 750b

   b. (2 pts) Convert 7bf4 from hexadecimal to binary 0111 1011 1111 0100

   c. (2 pts) Convert 0011 0010 from base 2 to base 10 50

   d. (2 pts) Convert 0100 0110 from base 2 to base 10 70

   e. (2 pts) Convert 1000 1110 1010 1001 from base 2 to hexadecimal 8ea9

   f. (2 pts) Convert 0111 0011 1110 1010 from binary to hexadecimal 73ea

   g. (2 pts) Convert 100 011 011 from base 2 to base 8 433

   h. (2 pts) Convert 0101 1001 0000 1101 from binary to base 16 590d

   i. (2 pts) Convert 80f7 from hexadecimal to base 2 1000 0000 1111 0111

   j. (2 pts) Convert 1110 from base 2 to decimal 14

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  Node c;
  int d;
  double e;
  char f;
  Node *g;
  int *h;
  double *p;
  char *q;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) g->next->next *Node **

b. (2 pts) e *double*

c. (2 pts) argv[0] *char **

d. (2 pts) argc *int*

e. (2 pts) argv[1][2] *char*

f. (2 pts) g->next *Node **

g. (2 pts) &f *char **

h. (2 pts) *p *double*

i. (2 pts) g->data *int*

j. (2 pts) h *int **

k. (2 pts) &p *double ***

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt fig mango grape
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of argv[2][1]? a

c. (2 pts) What is the value of argv[0][1]? /

d. (2 pts) What is the value of argv[1][1]? i

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 101 110 011 from binary to octal 563

   b. (2 pts) Convert 001 101 100 from binary to base 8 154

   c. (2 pts) Convert 10 from octal to base 2 001 000

   d. (2 pts) Convert 210 from base 10 to binary 1101 0010

   e. (2 pts) Convert 34 from base 8 to binary 011 100

   f. (2 pts) Convert 3 from base 8 to binary 011

   g. (2 pts) Convert 001 101 111 from binary to octal 157

   h. (2 pts) Convert 70 from base 8 to base 2 111 000

   i. (2 pts) Convert 4b0e from base 16 to binary 0100 1011 0000 1110

   j. (2 pts) Convert 101 010 100 from binary to octal 524

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  Node y;
  double z;
  int a;
  char b;
  Node *c;
  double *d;
  int *e;
  char *f;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) &c Node **

b. (2 pts) c->next->next Node *

c. (2 pts) *f char

d. (2 pts) c->data int

e. (2 pts) argc int

f. (2 pts) &a int *

g. (2 pts) argv[0] char *

h. (2 pts) y Node

i. (2 pts) c->next Node *

j. (2 pts) c Node *

k. (2 pts) argv[1][2] char

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt kiwi date
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of argv[2][0]? d

c. (2 pts) What is the value of argv[1][2]? w

d. (2 pts) What is the value of argv[0][4]? n

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 0001 0010 from base 2 to decimal 18

   b. (2 pts) Convert 1101 1000 1110 1000 from binary to hexadecimal d8e8

   c. (2 pts) Convert 110 001 100 from base 2 to octal 614

   d. (2 pts) Convert 43 from octal to binary 100 011

   e. (2 pts) Convert 226 from base 10 to base 2 1110 0010

   f. (2 pts) Convert 1111 from base 2 to decimal 15

   g. (2 pts) Convert 8978 from base 16 to base 2 1000 1001 0111 1000

   h. (2 pts) Convert 135 from decimal to base 2 1000 0111

   i. (2 pts) Convert 61 from decimal to base 2 0011 1101

   j. (2 pts) Convert 101 101 011 from base 2 to base 8 553

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double q;
  Node r;
  int s;
  char t;
  double *w;
  Node *x;
  int *y;
  char *z;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) argc *int*

b. (2 pts) &s *int **

c. (2 pts) x->data *int*

d. (2 pts) *z *char*

e. (2 pts) x->next *Node **

f. (2 pts) argv[0] *char **

g. (2 pts) z *char **

h. (2 pts) q *double*

i. (2 pts) x->next->next *Node **

j. (2 pts) argv[1][2] *char*

k. (2 pts) &z *char ***

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt grape banana
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of argv[2][1]? a

c. (2 pts) What is the value of argv[1][3]? p

d. (2 pts) What is the value of argv[0][4]? n

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 5726 from hexadecimal to base 2 0101 0111 0010 0110

   b. (2 pts) Convert 44 from base 8 to base 2 100 100

   c. (2 pts) Convert 101 101 000 from binary to octal 550

   d. (2 pts) Convert 0001 1001 1100 1100 from binary to base 16 19cc

   e. (2 pts) Convert 1100 0101 1011 1000 from base 2 to hexadecimal c5b8

   f. (2 pts) Convert 101 010 001 from base 2 to base 8 521

   g. (2 pts) Convert 3369 from hexadecimal to base 2 0011 0011 0110 1001

   h. (2 pts) Convert 1110 1101 1111 from base 2 to base 16 edf

   i. (2 pts) Convert 7 from base 10 to binary 0111

   j. (2 pts) Convert 24 from octal to base 2 010 100

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int f;
  double g;
  Node h;
  char p;
  int *q;
  double *r;
  Node *s;
  char *t;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) s *Node **

b. (2 pts) argc *int*

c. (2 pts) h *Node*

d. (2 pts) s->data *int*

e. (2 pts) &r *double ***

f. (2 pts) s->next *Node **

g. (2 pts) argv[1][2] *char*

h. (2 pts) *q *int*

i. (2 pts) &g *double **

j. (2 pts) s->next->next *Node **

k. (2 pts) argv[0] *char **

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt date grape lemon lime
```

a. (2 pts) What is the value of `argc` in this case? *5*

b. (2 pts) What is the value of argv[2][1]? *r*

c. (2 pts) What is the value of argv[0][0]? *.*

d. (2 pts) What is the value of argv[1][3]? *e*

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 136 from base 10 to base 2 1000 1000

   b. (2 pts) Convert 0110 0100 1001 0111 from base 2 to hexadecimal 6497

   c. (2 pts) Convert 1110 1010 0111 1000 from binary to hexadecimal ea78

   d. (2 pts) Convert 35 from octal to base 2 011 101

   e. (2 pts) Convert 6 from base 8 to binary 110

   f. (2 pts) Convert 219 from decimal to binary 1101 1011

   g. (2 pts) Convert 3597 from base 16 to base 2 0011 0101 1001 0111

   h. (2 pts) Convert 0111 0111 1100 1110 from binary to base 16 77ce

   i. (2 pts) Convert 169 from decimal to base 2 1010 1001

   j. (2 pts) Convert 011 111 011 from binary to base 8 373

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  Node g;
  int h;
  double p;
  char q;
  Node *r;
  int *s;
  double *t;
  char *w;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) r->next->next Node *

b. (2 pts) &t double **

c. (2 pts) r->data int

d. (2 pts) &g Node *

e. (2 pts) argc int

f. (2 pts) q char

g. (2 pts) s int *

h. (2 pts) r->next Node *

i. (2 pts) argv[0] char *

j. (2 pts) *r Node

k. (2 pts) argv[1][2] char

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt grape banana fig
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of argv[1][1]? r

c. (2 pts) What is the value of argv[2][5]? a

d. (2 pts) What is the value of argv[0][2]? r

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 1100 1101 0110 0111 from base 2 to hexadecimal cd67

   b. (2 pts) Convert 0001 1110 from binary to base 10 30

   c. (2 pts) Convert 216 from decimal to binary 1101 1000

   d. (2 pts) Convert 0 from octal to binary 000

   e. (2 pts) Convert 111 111 110 from base 2 to base 8 776

   f. (2 pts) Convert 117 from decimal to binary 0111 0101

   g. (2 pts) Convert df88 from hexadecimal to base 2 1101 1111 1000 1000

   h. (2 pts) Convert 1111 1111 from binary to base 10 255

   i. (2 pts) Convert 736a from base 16 to binary 0111 0011 0110 1010

   j. (2 pts) Convert 110 011 from base 2 to base 8 63

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double t;
  Node w;
  int x;
  char y;
  double *z;
  Node *a;
  int *b;
  char *c;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) z double *

b. (2 pts) a->next Node *

c. (2 pts) argv[0] char *

d. (2 pts) &y char *

e. (2 pts) argv[1][2] char

f. (2 pts) a->next->next Node *

g. (2 pts) *a Node

h. (2 pts) &a Node **

i. (2 pts) a->data int

j. (2 pts) argc int

k. (2 pts) x int

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt fig cherry
```

a. (2 pts) What is the value of `argc` in this case? *3*

b. (2 pts) What is the value of argv[1][0]? *f*

c. (2 pts) What is the value of argv[2][3]? *r*

d. (2 pts) What is the value of argv[0][0]? *.*

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 226 from base 10 to base 2 1110 0010

   b. (2 pts) Convert 1101 1101 1101 1100 from base 2 to hexadecimal dddc

   c. (2 pts) Convert 010 110 010 from base 2 to octal 262

   d. (2 pts) Convert 80 from decimal to binary 0101 0000

   e. (2 pts) Convert 43 from base 8 to binary 100 011

   f. (2 pts) Convert d542 from base 16 to base 2 1101 0101 0100 0010

   g. (2 pts) Convert 011 111 101 from base 2 to base 8 375

   h. (2 pts) Convert 235 from base 10 to binary 1110 1011

   i. (2 pts) Convert 100 100 011 from binary to octal 443

   j. (2 pts) Convert 101 110 000 from base 2 to octal 560

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double h;
  Node p;
  int q;
  char r;
  double *s;
  Node *t;
  int *w;
  char *x;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) x *char **

b. (2 pts) q *int*

c. (2 pts) *t *Node*

d. (2 pts) &w *int ***

e. (2 pts) argv[1][2] *char*

f. (2 pts) &p *Node **

g. (2 pts) t->data *int*

h. (2 pts) argc *int*

i. (2 pts) t->next *Node **

j. (2 pts) argv[0] *char **

k. (2 pts) t->next->next *Node **

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt lime grape
```

a. (2 pts) What is the value of `argc` in this case? *3*

b. (2 pts) What is the value of argv[1][2]? *m*

c. (2 pts) What is the value of argv[2][2]? *a*

d. (2 pts) What is the value of argv[0][0]? *.*

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 0010 0111 0100 0001 from base 2 to base 16 2741

   b. (2 pts) Convert 1001 1000 from base 2 to base 10 152

   c. (2 pts) Convert 0100 0111 0100 0110 from binary to hexadecimal 4746

   d. (2 pts) Convert 220 from decimal to base 2 1101 1100

   e. (2 pts) Convert 011 100 001 from base 2 to base 8 341

   f. (2 pts) Convert 111 from base 10 to base 2 0110 1111

   g. (2 pts) Convert 001 010 001 from base 2 to base 8 121

   h. (2 pts) Convert 0111 0011 0101 0000 from binary to hexadecimal 7350

   i. (2 pts) Convert 0101 1011 from base 2 to base 10 91

   j. (2 pts) Convert 25 from octal to binary 010 101

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double e;
  Node f;
  int g;
  char h;
  double *p;
  Node *q;
  int *r;
  char *s;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) r *int **

b. (2 pts) e *double*

c. (2 pts) *r *int*

d. (2 pts) q->next *Node **

e. (2 pts) argc *int*

f. (2 pts) argv[1][2] *char*

g. (2 pts) &p *double ***

h. (2 pts) q->next->next *Node **

i. (2 pts) q->data *int*

j. (2 pts) argv[0] *char **

k. (2 pts) &f *Node **

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt lemon mango
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of argv[2][4]? o

c. (2 pts) What is the value of argv[1][2]? m

d. (2 pts) What is the value of argv[0][5]? I

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 58a3 from hexadecimal to binary 0101 1000 1010 0011

   b. (2 pts) Convert 011 010 011 from binary to octal 323

   c. (2 pts) Convert 0111 1101 from binary to decimal 125

   d. (2 pts) Convert 38cb from hexadecimal to binary 0011 1000 1100 1011

   e. (2 pts) Convert 1100 0111 from base 2 to base 10 199

   f. (2 pts) Convert 50 from octal to binary 101 000

   g. (2 pts) Convert 2acf from base 16 to binary 0010 1010 1100 1111

   h. (2 pts) Convert 1101 1100 0011 1111 from binary to hexadecimal dc3f

   i. (2 pts) Convert 1111 1101 from binary to decimal 253

   j. (2 pts) Convert 100 000 000 from binary to octal 400

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int x;
  double y;
  Node z;
  char a;
  int *b;
  double *c;
  Node *d;
  char *e;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) d->data *int*

b. (2 pts) argc *int*

c. (2 pts) c *double **

d. (2 pts) d->next *Node **

e. (2 pts) argv[1][2] *char*

f. (2 pts) y *double*

g. (2 pts) argv[0] *char **

h. (2 pts) d->next->next *Node **

i. (2 pts) &a *char **

j. (2 pts) &d *Node ***

k. (2 pts) *d *Node*

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt apple kiwi lemon fig
```

a. (2 pts) What is the value of `argc` in this case? *5*

b. (2 pts) What is the value of argv[1][4]? *e*

c. (2 pts) What is the value of argv[0][2]? *r*

d. (2 pts) What is the value of argv[2][2]? *w*

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 47 from octal to binary 100 111

   b. (2 pts) Convert 23b3 from hexadecimal to base 2 0010 0011 1011 0011

   c. (2 pts) Convert 32 from octal to binary 011 010

   d. (2 pts) Convert 1100 0100 1011 0101 from binary to base 16 c4b5

   e. (2 pts) Convert aa5f from base 16 to binary 1010 1010 0101 1111

   f. (2 pts) Convert 0011 1011 0110 0101 from binary to hexadecimal 3b65

   g. (2 pts) Convert 65 from octal to base 2 110 101

   h. (2 pts) Convert 0110 0011 from base 2 to decimal 99

   i. (2 pts) Convert 199 from base 10 to base 2 1100 0111

   j. (2 pts) Convert 111 000 from base 2 to octal 70

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int s;
  Node t;
  double w;
  char x;
  int *y;
  Node *z;
  double *a;
  char *b;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) &w double *

b. (2 pts) z->next Node *

c. (2 pts) argc int

d. (2 pts) argv[0] char *

e. (2 pts) z->data int

f. (2 pts) z->next->next Node *

g. (2 pts) &y int **

h. (2 pts) y int *

i. (2 pts) s int

j. (2 pts) *b char

k. (2 pts) argv[1][2] char

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt cherry date apple
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of argv[0][5]? I

c. (2 pts) What is the value of argv[2][1]? a

d. (2 pts) What is the value of argv[1][5]? y

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 111 101 011 from binary to octal 753

   b. (2 pts) Convert 61 from octal to binary 110 001

   c. (2 pts) Convert 100 001 from binary to base 8 41

   d. (2 pts) Convert 1000 0000 from binary to base 10 128

   e. (2 pts) Convert 0001 1010 from base 2 to base 10 26

   f. (2 pts) Convert 60 from decimal to binary 0011 1100

   g. (2 pts) Convert 001 001 100 from binary to octal 114

   h. (2 pts) Convert 10 from base 10 to binary 1010

   i. (2 pts) Convert 101 110 100 from base 2 to base 8 564

   j. (2 pts) Convert 11 from octal to base 2 001 001

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  Node e;
  double f;
  int g;
  char h;
  Node *p;
  double *q;
  int *r;
  char *s;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) p->next->next *Node **

b. (2 pts) p->data *int*

c. (2 pts) &s *char ***

d. (2 pts) argv[0] *char **

e. (2 pts) argv[1][2] *char*

f. (2 pts) *s *char*

g. (2 pts) argc *int*

h. (2 pts) h *char*

i. (2 pts) &g *int **

j. (2 pts) p->next *Node **

k. (2 pts) s *char **

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt banana lime
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of argv[0][0]? .

c. (2 pts) What is the value of argv[2][0]? l

d. (2 pts) What is the value of argv[1][1]? a

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 58 from base 10 to base 2 *0011 1010*

   b. (2 pts) Convert 128 from base 10 to base 2 *1000 0000*

   c. (2 pts) Convert 1111 1110 from binary to base 10 *254*

   d. (2 pts) Convert 011 000 from base 2 to octal *30*

   e. (2 pts) Convert fe23 from base 16 to base 2 *1111 1110 0010 0011*

   f. (2 pts) Convert 1101 0110 from base 2 to base 10 *214*

   g. (2 pts) Convert 208 from base 10 to binary *1101 0000*

   h. (2 pts) Convert 1001 0010 0001 0001 from base 2 to hexadecimal *9211*

   i. (2 pts) Convert 100 001 000 from binary to octal *410*

   j. (2 pts) Convert 60 from base 8 to base 2 *110 000*

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double a;
  Node b;
  int c;
  char d;
  double *e;
  Node *f;
  int *g;
  char *h;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) f->next *Node **

b. (2 pts) *e *double*

c. (2 pts) &b *Node **

d. (2 pts) f->next->next *Node **

e. (2 pts) &f *Node ***

f. (2 pts) argv[1][2] *char*

g. (2 pts) f->data *int*

h. (2 pts) argc *int*

i. (2 pts) g *int **

j. (2 pts) b *Node*

k. (2 pts) argv[0] *char **

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt guava lime
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of argv[1][1]? u

c. (2 pts) What is the value of argv[0][1]? /

d. (2 pts) What is the value of argv[2][0]? l

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 0110 1100 from base 2 to decimal 108

   b. (2 pts) Convert 0101 0010 from base 2 to decimal 82

   c. (2 pts) Convert 0011 0101 0010 0110 from binary to base 16 3526

   d. (2 pts) Convert 0110 1000 from binary to decimal 104

   e. (2 pts) Convert 84 from base 10 to binary 0101 0100

   f. (2 pts) Convert 8fb from base 16 to base 2 1000 1111 1011

   g. (2 pts) Convert 110 100 101 from binary to base 8 645

   h. (2 pts) Convert 76 from octal to binary 111 110

   i. (2 pts) Convert 001 001 011 from base 2 to octal 113

   j. (2 pts) Convert 74 from base 8 to base 2 111 100

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double r;
  int s;
  Node t;
  char w;
  double *x;
  int *y;
  Node *z;
  char *a;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) y *int **

b. (2 pts) *x *double*

c. (2 pts) &r *double **

d. (2 pts) argv[0] *char **

e. (2 pts) w *char*

f. (2 pts) &z *Node ***

g. (2 pts) z->next *Node **

h. (2 pts) z->data *int*

i. (2 pts) z->next->next *Node **

j. (2 pts) argv[1][2] *char*

k. (2 pts) argc *int*

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt kiwi guava lemon lime
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of argv[2][1]? u

c. (2 pts) What is the value of argv[0][0]? .

d. (2 pts) What is the value of argv[1][3]? i

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert b0ff from hexadecimal to binary 1011 0000 1111 1111

   b. (2 pts) Convert 1100 0101 0110 from binary to base 16 c56

   c. (2 pts) Convert 35 from base 10 to base 2 0010 0011

   d. (2 pts) Convert 1111 0100 from binary to decimal 244

   e. (2 pts) Convert 001 101 111 from base 2 to octal 157

   f. (2 pts) Convert a2d8 from hexadecimal to binary 1010 0010 1101 1000

   g. (2 pts) Convert 011 111 001 from base 2 to octal 371

   h. (2 pts) Convert 1000 0010 from binary to decimal 130

   i. (2 pts) Convert 1110 1111 from binary to base 10 239

   j. (2 pts) Convert 42 from octal to base 2 100 010

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int h;
  Node p;
  double q;
  char r;
  int *s;
  Node *t;
  double *w;
  char *x;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) s *int **

b. (2 pts) argv[0] *char **

c. (2 pts) &r *char **

d. (2 pts) argc *int*

e. (2 pts) p *Node*

f. (2 pts) t->next *Node **

g. (2 pts) argv[1][2] *char*

h. (2 pts) &s *int ***

i. (2 pts) *t *Node*

j. (2 pts) t->next->next *Node **

k. (2 pts) t->data *int*

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt lemon lime date
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of argv[0][3]? u

c. (2 pts) What is the value of argv[2][2]? m

d. (2 pts) What is the value of argv[1][3]? o

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 95c from hexadecimal to binary 1001 0101 1100

   b. (2 pts) Convert af22 from hexadecimal to binary 1010 1111 0010 0010

   c. (2 pts) Convert c869 from hexadecimal to binary 1100 1000 0110 1001

   d. (2 pts) Convert 53 from base 8 to base 2 101 011

   e. (2 pts) Convert 168 from base 10 to base 2 1010 1000

   f. (2 pts) Convert 1010 0100 0010 1001 from binary to hexadecimal a429

   g. (2 pts) Convert ce35 from hexadecimal to base 2 1100 1110 0011 0101

   h. (2 pts) Convert 0010 1001 0010 0010 from binary to base 16 2922

   i. (2 pts) Convert 1110 0010 from base 2 to base 10 226

   j. (2 pts) Convert 151 from base 10 to binary 1001 0111

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  Node a;
  double b;
  int c;
  char d;
  Node *e;
  double *f;
  int *g;
  char *h;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) e->data *int*

b. (2 pts) argv[1][2] *char*

c. (2 pts) a *Node*

d. (2 pts) *f *double*

e. (2 pts) argv[0] *char **

f. (2 pts) g *int **

g. (2 pts) e->next->next *Node **

h. (2 pts) &c *int **

i. (2 pts) e->next *Node **

j. (2 pts) &h *char ***

k. (2 pts) argc *int*

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt kiwi banana
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of argv[2][1]? a

c. (2 pts) What is the value of argv[1][1]? i

d. (2 pts) What is the value of argv[0][5]? I

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 010 011 100 from binary to octal 234

   b. (2 pts) Convert 0110 1001 0100 1011 from binary to hexadecimal 694b

   c. (2 pts) Convert 182 from base 10 to binary 1011 0110

   d. (2 pts) Convert 3bdb from base 16 to binary 0011 1011 1101 1011

   e. (2 pts) Convert 1000 1011 0101 0000 from base 2 to hexadecimal 8b50

   f. (2 pts) Convert 3e06 from hexadecimal to binary 0011 1110 0000 0110

   g. (2 pts) Convert 7826 from base 16 to base 2 0111 1000 0010 0110

   h. (2 pts) Convert 54 from base 8 to base 2 101 100

   i. (2 pts) Convert 101 011 000 from binary to octal 530

   j. (2 pts) Convert 51 from decimal to binary 0011 0011

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int x;
  double y;
  Node z;
  char a;
  int *b;
  double *c;
  Node *d;
  char *e;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) &d Node **

b. (2 pts) &z Node *

c. (2 pts) d->next->next Node *

d. (2 pts) argc int

e. (2 pts) d->data int

f. (2 pts) c double *

g. (2 pts) a char

h. (2 pts) argv[1][2] char

i. (2 pts) argv[0] char *

j. (2 pts) *d Node

k. (2 pts) d->next Node *

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt apple grape fig cherry
```

a. (2 pts) What is the value of `argc` in this case? *5*

b. (2 pts) What is the value of argv[1][2]? *p*

c. (2 pts) What is the value of argv[0][6]? *t*

d. (2 pts) What is the value of argv[2][3]? *p*

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 0111 1111 1001 1101 from base 2 to base 16 7f9d

   b. (2 pts) Convert 16 from base 8 to binary 001 110

   c. (2 pts) Convert 73 from base 8 to base 2 111 011

   d. (2 pts) Convert 100 110 000 from base 2 to base 8 460

   e. (2 pts) Convert 70 from base 8 to binary 111 000

   f. (2 pts) Convert 011 100 000 from base 2 to base 8 340

   g. (2 pts) Convert 122 from decimal to base 2 0111 1010

   h. (2 pts) Convert 6 from base 8 to binary 110

   i. (2 pts) Convert 0100 1110 from base 2 to base 10 78

   j. (2 pts) Convert 27 from octal to base 2 010 111

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  Node x;
  int y;
  double z;
  char a;
  Node *b;
  int *c;
  double *d;
  char *e;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) argv[1][2] *char*

b. (2 pts) *d *double*

c. (2 pts) b->next->next *Node **

d. (2 pts) x *Node*

e. (2 pts) &e *char ***

f. (2 pts) argv[0] *char **

g. (2 pts) argc *int*

h. (2 pts) b *Node **

i. (2 pts) b->next *Node **

j. (2 pts) &x *Node **

k. (2 pts) b->data *int*

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt cherry grape kiwi
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of argv[0][2]? r

c. (2 pts) What is the value of argv[1][0]? c

d. (2 pts) What is the value of argv[2][3]? p

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 1100 0100 from base 2 to decimal 196

   b. (2 pts) Convert 111 101 001 from binary to base 8 751

   c. (2 pts) Convert da88 from hexadecimal to binary 1101 1010 1000 1000

   d. (2 pts) Convert 0010 0100 0000 0110 from base 2 to hexadecimal 2406

   e. (2 pts) Convert 110 001 001 from base 2 to base 8 611

   f. (2 pts) Convert 2 from octal to binary 010

   g. (2 pts) Convert 2445 from base 16 to base 2 0010 0100 0100 0101

   h. (2 pts) Convert 1010 0001 from binary to decimal 161

   i. (2 pts) Convert 110 000 from binary to octal 60

   j. (2 pts) Convert 251 from decimal to binary 1111 1011

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double d;
  Node e;
  int f;
  char g;
  double *h;
  Node *p;
  int *q;
  char *r;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) &p Node **

b. (2 pts) p->next->next Node *

c. (2 pts) argc int

d. (2 pts) *r char

e. (2 pts) p->data int

f. (2 pts) argv[1][2] char

g. (2 pts) p->next Node *

h. (2 pts) argv[0] char *

i. (2 pts) f int

j. (2 pts) r char *

k. (2 pts) &g char *

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt fig cherry
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of argv[2][0]? c

c. (2 pts) What is the value of argv[0][0]? .

d. (2 pts) What is the value of argv[1][2]? g

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 0001 1100 from base 2 to base 10 28

   b. (2 pts) Convert 1001 0111 1100 0101 from binary to base 16 97c5

   c. (2 pts) Convert 0111 1111 1110 1010 from base 2 to base 16 7fea

   d. (2 pts) Convert 223 from base 10 to binary 1101 1111

   e. (2 pts) Convert 3532 from base 16 to binary 0011 0101 0011 0010

   f. (2 pts) Convert b9c from hexadecimal to binary 1011 1001 1100

   g. (2 pts) Convert 35 from octal to binary 011 101

   h. (2 pts) Convert 0100 0111 1110 0100 from binary to base 16 47e4

   i. (2 pts) Convert 010 101 from base 2 to base 8 25

   j. (2 pts) Convert 0111 from binary to decimal 7

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double e;
  Node f;
  int g;
  char h;
  double *p;
  Node *q;
  int *r;
  char *s;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) q->data *int*

b. (2 pts) f *Node*

c. (2 pts) r *int **

d. (2 pts) argv[0] *char **

e. (2 pts) q->next *Node **

f. (2 pts) &p *double ***

g. (2 pts) *r *int*

h. (2 pts) argc *int*

i. (2 pts) &h *char **

j. (2 pts) q->next->next *Node **

k. (2 pts) argv[1][2] *char*

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt lime apple
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of argv[2][2]? p

c. (2 pts) What is the value of argv[1][2]? m

d. (2 pts) What is the value of argv[0][1]? /

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 0110 0001 1011 1000 from base 2 to hexadecimal 61b8

   b. (2 pts) Convert 0101 0001 from binary to decimal 81

   c. (2 pts) Convert 0110 1101 1100 1011 from binary to base 16 6dcb

   d. (2 pts) Convert 107 from base 10 to base 2 0110 1011

   e. (2 pts) Convert 0001 1000 0111 1110 from base 2 to hexadecimal 187e

   f. (2 pts) Convert 165 from decimal to binary 1010 0101

   g. (2 pts) Convert 7 from base 8 to binary 111

   h. (2 pts) Convert 63 from octal to base 2 110 011

   i. (2 pts) Convert 65 from octal to binary 110 101

   j. (2 pts) Convert 1010 0011 from base 2 to base 10 163

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double r;
  int s;
  Node t;
  char w;
  double *x;
  int *y;
  Node *z;
  char *a;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) &z *Node **

b. (2 pts) r *double*

c. (2 pts) z->next *Node **

d. (2 pts) argc *int*

e. (2 pts) z->data *int*

f. (2 pts) *x *double*

g. (2 pts) argv[0] *char **

h. (2 pts) &t *Node **

i. (2 pts) y *int **

j. (2 pts) argv[1][2] *char*

k. (2 pts) z->next->next *Node **

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt date guava grape lemon
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of argv[1][0]? d

c. (2 pts) What is the value of argv[2][2]? a

d. (2 pts) What is the value of argv[0][6]? t

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 147 from decimal to base 2 *1001 0011*

   b. (2 pts) Convert 001 000 110 from base 2 to base 8 *106*

   c. (2 pts) Convert 164 from decimal to binary *1010 0100*

   d. (2 pts) Convert c7c5 from base 16 to binary *1100 0111 1100 0101*

   e. (2 pts) Convert 33 from base 8 to base 2 *011 011*

   f. (2 pts) Convert 65 from base 8 to base 2 *110 101*

   g. (2 pts) Convert 2216 from hexadecimal to binary *0010 0010 0001 0110*

   h. (2 pts) Convert 0011 1000 from base 2 to base 10 *56*

   i. (2 pts) Convert 0111 0110 from base 2 to base 10 *118*

   j. (2 pts) Convert 207 from base 10 to binary *1100 1111*

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int s;
  Node t;
  double w;
  char x;
  int *y;
  Node *z;
  double *a;
  char *b;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) &y int **

b. (2 pts) z->next Node *

c. (2 pts) z->next->next Node *

d. (2 pts) y int *

e. (2 pts) argv[0] char *

f. (2 pts) &t Node *

g. (2 pts) argv[1][2] char

h. (2 pts) z->data int

i. (2 pts) argc int

j. (2 pts) *y int

k. (2 pts) t Node

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt apple mango fig
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of argv[0][0]? .

c. (2 pts) What is the value of argv[1][4]? e

d. (2 pts) What is the value of argv[2][2]? n

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 65 from base 8 to base 2 110 101

   b. (2 pts) Convert 1101 1101 from base 2 to decimal 221

   c. (2 pts) Convert 100 100 100 from binary to octal 444

   d. (2 pts) Convert 0101 0011 1010 1111 from base 2 to base 16 53af

   e. (2 pts) Convert 0101 0001 from base 2 to base 10 81

   f. (2 pts) Convert 0111 0001 1100 0000 from base 2 to base 16 71c0

   g. (2 pts) Convert cc07 from base 16 to binary 1100 1100 0000 0111

   h. (2 pts) Convert 192 from base 10 to base 2 1100 0000

   i. (2 pts) Convert 20 from octal to binary 010 000

   j. (2 pts) Convert 107 from base 10 to binary 0110 1011

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double h;
  Node p;
  int q;
  char r;
  double *s;
  Node *t;
  int *w;
  char *x;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) argc *int*

b. (2 pts) t->data *int*

c. (2 pts) x *char **

d. (2 pts) &h *double **

e. (2 pts) *t *Node*

f. (2 pts) t->next *Node **

g. (2 pts) argv[1][2] *char*

h. (2 pts) r *char*

i. (2 pts) t->next->next *Node **

j. (2 pts) argv[0] *char **

k. (2 pts) &w *int ***

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt lime mango
```

a. (2 pts) What is the value of `argc` in this case? *3*

b. (2 pts) What is the value of argv[2][1]? *a*

c. (2 pts) What is the value of argv[0][3]? *u*

d. (2 pts) What is the value of argv[1][2]? *m*

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 45 from base 8 to binary 100 101

   b. (2 pts) Convert 59 from decimal to binary 0011 1011

   c. (2 pts) Convert 0111 1011 0110 0000 from base 2 to base 16 7b60

   d. (2 pts) Convert 44 from base 8 to binary 100 100

   e. (2 pts) Convert 010 110 001 from base 2 to octal 261

   f. (2 pts) Convert 0011 0111 from binary to decimal 55

   g. (2 pts) Convert dfa6 from base 16 to binary 1101 1111 1010 0110

   h. (2 pts) Convert 34 from base 8 to binary 011 100

   i. (2 pts) Convert 1111 0000 from base 2 to decimal 240

   j. (2 pts) Convert f995 from hexadecimal to base 2 1111 1001 1001 0101

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double x;
  Node y;
  int z;
  char a;
  double *b;
  Node *c;
  int *d;
  char *e;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) *c Node

b. (2 pts) c->data int

c. (2 pts) argv[0] char *

d. (2 pts) &a char *

e. (2 pts) argc int

f. (2 pts) b double *

g. (2 pts) argv[1][2] char

h. (2 pts) &e char **

i. (2 pts) c->next Node *

j. (2 pts) c->next->next Node *

k. (2 pts) y Node

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt lime lemon
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of argv[2][3]? o

c. (2 pts) What is the value of argv[1][2]? m

d. (2 pts) What is the value of argv[0][3]? u

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 1101 1001 from binary to base 10 217

   b. (2 pts) Convert 75 from octal to base 2 111 101

   c. (2 pts) Convert 6940 from base 16 to base 2 0110 1001 0100 0000

   d. (2 pts) Convert 7 from base 8 to base 2 111

   e. (2 pts) Convert 0011 1100 from base 2 to base 10 60

   f. (2 pts) Convert 110 100 010 from binary to octal 642

   g. (2 pts) Convert 100 010 011 from binary to base 8 423

   h. (2 pts) Convert 1111 1000 from base 2 to decimal 248

   i. (2 pts) Convert 186 from decimal to base 2 1011 1010

   j. (2 pts) Convert 149 from decimal to base 2 1001 0101

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int s;
  double t;
  Node w;
  char x;
  int *y;
  double *z;
  Node *a;
  char *b;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) a->next->next *Node **

b. (2 pts) argc *int*

c. (2 pts) a->next *Node **

d. (2 pts) &z *double ***

e. (2 pts) b *char **

f. (2 pts) a->data *int*

g. (2 pts) argv[0] *char **

h. (2 pts) s *int*

i. (2 pts) argv[1][2] *char*

j. (2 pts) &w *Node **

k. (2 pts) *b *char*

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt cherry guava fig grape
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of argv[1][2]? e

c. (2 pts) What is the value of argv[2][1]? u

d. (2 pts) What is the value of argv[0][1]? /

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 010 101 from base 2 to octal 25

   b. (2 pts) Convert 1100 0110 1111 1110 from base 2 to hexadecimal c6fe

   c. (2 pts) Convert 100 111 111 from base 2 to base 8 477

   d. (2 pts) Convert 011 110 100 from base 2 to octal 364

   e. (2 pts) Convert 146 from base 10 to base 2 1001 0010

   f. (2 pts) Convert 3b9 from base 16 to base 2 0011 1011 1001

   g. (2 pts) Convert 8bc5 from base 16 to binary 1000 1011 1100 0101

   h. (2 pts) Convert 97 from decimal to base 2 0110 0001

   i. (2 pts) Convert 92 from base 10 to base 2 0101 1100

   j. (2 pts) Convert c1c0 from hexadecimal to base 2 1100 0001 1100 0000

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double e;
  int f;
  Node g;
  char h;
  double *p;
  int *q;
  Node *r;
  char *s;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) s *char **

b. (2 pts) r->next->next *Node **

c. (2 pts) *r *Node*

d. (2 pts) argv[1][2] *char*

e. (2 pts) argv[0] *char **

f. (2 pts) f *int*

g. (2 pts) &s *char ***

h. (2 pts) r->next *Node **

i. (2 pts) &f *int **

j. (2 pts) r->data *int*

k. (2 pts) argc *int*

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt grape date apple lemon
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of argv[1][0]? g

c. (2 pts) What is the value of argv[0][5]? I

d. (2 pts) What is the value of argv[2][2]? t

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 79 from base 10 to binary 0100 1111

   b. (2 pts) Convert 40 from base 8 to base 2 100 000

   c. (2 pts) Convert 100 011 010 from binary to base 8 432

   d. (2 pts) Convert 0110 0001 0010 from base 2 to base 16 612

   e. (2 pts) Convert 0111 0101 1010 1001 from base 2 to hexadecimal 75a9

   f. (2 pts) Convert 157 from base 10 to binary 1001 1101

   g. (2 pts) Convert 35b6 from hexadecimal to binary 0011 0101 1011 0110

   h. (2 pts) Convert 232 from base 10 to base 2 1110 1000

   i. (2 pts) Convert 261b from hexadecimal to base 2 0010 0110 0001 1011

   j. (2 pts) Convert 5daa from base 16 to base 2 0101 1101 1010 1010

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  Node b;
  int c;
  double d;
  char e;
  Node *f;
  int *g;
  double *h;
  char *p;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) argc int

b. (2 pts) e char

c. (2 pts) *f Node

d. (2 pts) &b Node *

e. (2 pts) &g int **

f. (2 pts) f->next->next Node *

g. (2 pts) argv[0] char *

h. (2 pts) argv[1][2] char

i. (2 pts) h double *

j. (2 pts) f->data int

k. (2 pts) f->next Node *

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt date grape banana
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of argv[2][2]? a

c. (2 pts) What is the value of argv[0][0]? .

d. (2 pts) What is the value of argv[1][2]? t

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 167 from base 10 to binary 1010 0111

   b. (2 pts) Convert 23f3 from base 16 to base 2 0010 0011 1111 0011

   c. (2 pts) Convert 14 from base 8 to binary 001 100

   d. (2 pts) Convert 193 from base 10 to binary 1100 0001

   e. (2 pts) Convert 230 from base 10 to base 2 1110 0110

   f. (2 pts) Convert 47 from base 8 to binary 100 111

   g. (2 pts) Convert 100 001 110 from base 2 to octal 416

   h. (2 pts) Convert 43 from octal to binary 100 011

   i. (2 pts) Convert 188f from hexadecimal to binary 0001 1000 1000 1111

   j. (2 pts) Convert 105 from base 10 to binary 0110 1001

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double s;
  Node t;
  int w;
  char x;
  double *y;
  Node *z;
  int *a;
  char *b;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) y double *

b. (2 pts) argv[1][2] char

c. (2 pts) z->next Node *

d. (2 pts) x char

e. (2 pts) z->next->next Node *

f. (2 pts) *b char

g. (2 pts) &y double **

h. (2 pts) argv[0] char *

i. (2 pts) z->data int

j. (2 pts) argc int

k. (2 pts) &s double *

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt date guava
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of argv[2][0]? g

c. (2 pts) What is the value of argv[1][3]? e

d. (2 pts) What is the value of argv[0][5]? I

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 73 from base 8 to binary 111 011

   b. (2 pts) Convert 110 111 100 from binary to octal 674

   c. (2 pts) Convert 10 from octal to binary 001 000

   d. (2 pts) Convert 77 from base 10 to binary 0100 1101

   e. (2 pts) Convert 1100 1001 0110 1101 from base 2 to hexadecimal c96d

   f. (2 pts) Convert 16 from octal to binary 001 110

   g. (2 pts) Convert 001 100 010 from binary to octal 142

   h. (2 pts) Convert 0001 0111 from base 2 to decimal 23

   i. (2 pts) Convert 1110 0010 from base 2 to base 10 226

   j. (2 pts) Convert 5 from base 10 to binary 0101

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double h;
  int p;
  Node q;
  char r;
  double *s;
  int *t;
  Node *w;
  char *x;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) x *char **

b. (2 pts) w->data *int*

c. (2 pts) w->next *Node **

d. (2 pts) p *int*

e. (2 pts) argv[1][2] *char*

f. (2 pts) w->next->next *Node **

g. (2 pts) argc *int*

h. (2 pts) &r *char **

i. (2 pts) &w *Node ***

j. (2 pts) *s *double*

k. (2 pts) argv[0] *char **

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt date banana fig kiwi
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of argv[1][0]? d

c. (2 pts) What is the value of argv[2][0]? b

d. (2 pts) What is the value of argv[0][2]? r

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 1e2b from hexadecimal to base 2 *0001 1110 0010 1011*

   b. (2 pts) Convert 53 from octal to base 2 *101 011*

   c. (2 pts) Convert 5720 from hexadecimal to base 2 *0101 0111 0010 0000*

   d. (2 pts) Convert a9d0 from base 16 to binary *1010 1001 1101 0000*

   e. (2 pts) Convert 1f8b from base 16 to binary *0001 1111 1000 1011*

   f. (2 pts) Convert 0110 1011 from binary to base 10 *107*

   g. (2 pts) Convert 001 100 111 from base 2 to base 8 *147*

   h. (2 pts) Convert 127 from decimal to binary *0111 1111*

   i. (2 pts) Convert 1000 0100 from base 2 to decimal *132*

   j. (2 pts) Convert 49 from base 10 to base 2 *0011 0001*

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int p;
  double q;
  Node r;
  char s;
  int *t;
  double *w;
  Node *x;
  char *y;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) x->data int

b. (2 pts) &r Node *

c. (2 pts) r Node

d. (2 pts) x->next->next Node *

e. (2 pts) *t int

f. (2 pts) x->next Node *

g. (2 pts) &t int **

h. (2 pts) argc int

i. (2 pts) argv[0] char *

j. (2 pts) x Node *

k. (2 pts) argv[1][2] char

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt cherry fig lemon banana
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of argv[0][6]? t

c. (2 pts) What is the value of argv[1][4]? r

d. (2 pts) What is the value of argv[2][2]? g

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 011 000 110 from binary to octal 306

   b. (2 pts) Convert 105 from base 10 to binary 0110 1001

   c. (2 pts) Convert 0100 0101 from binary to decimal 69

   d. (2 pts) Convert 0011 0101 1011 1011 from base 2 to base 16 35bb

   e. (2 pts) Convert 0 from base 8 to binary 000

   f. (2 pts) Convert 001 010 from base 2 to base 8 12

   g. (2 pts) Convert 110 111 010 from binary to base 8 672

   h. (2 pts) Convert 0111 1010 1011 from binary to hexadecimal 7ab

   i. (2 pts) Convert 78 from base 10 to base 2 0100 1110

   j. (2 pts) Convert 205 from decimal to binary 1100 1101

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int x;
  Node y;
  double z;
  char a;
  int *b;
  Node *c;
  double *d;
  char *e;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) argv[1][2] *char*

b. (2 pts) &y *Node **

c. (2 pts) x *int*

d. (2 pts) c->next *Node **

e. (2 pts) *c *Node*

f. (2 pts) c->data *int*

g. (2 pts) argc *int*

h. (2 pts) c *Node **

i. (2 pts) argv[0] *char **

j. (2 pts) &d *double ***

k. (2 pts) c->next->next *Node **

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt cherry guava fig
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of argv[2][3]? v

c. (2 pts) What is the value of argv[0][5]? I

d. (2 pts) What is the value of argv[1][3]? r

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 101 110 110 from binary to octal 566

   b. (2 pts) Convert 1100 from base 2 to base 10 12

   c. (2 pts) Convert 234 from decimal to base 2 1110 1010

   d. (2 pts) Convert 1111 0001 from base 2 to base 10 241

   e. (2 pts) Convert 734f from base 16 to base 2 0111 0011 0100 1111

   f. (2 pts) Convert 6 from base 10 to base 2 0110

   g. (2 pts) Convert 2f2b from base 16 to base 2 0010 1111 0010 1011

   h. (2 pts) Convert 1010 1110 from binary to decimal 174

   i. (2 pts) Convert 64 from decimal to binary 0100 0000

   j. (2 pts) Convert 1101 1000 from binary to base 10 216

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  Node h;
  double p;
  int q;
  char r;
  Node *s;
  double *t;
  int *w;
  char *x;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) s->next->next *Node **

b. (2 pts) *t *double*

c. (2 pts) &t *double***

d. (2 pts) s *Node **

e. (2 pts) argc *int*

f. (2 pts) s->data *int*

g. (2 pts) &h *Node **

h. (2 pts) h *Node*

i. (2 pts) s->next *Node **

j. (2 pts) argv[1][2] *char*

k. (2 pts) argv[0] *char **

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt grape mango
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of argv[0][5]? I

c. (2 pts) What is the value of argv[2][0]? m

d. (2 pts) What is the value of argv[1][0]? g

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 46 from hexadecimal to binary 0100 0110

   b. (2 pts) Convert 198 from decimal to binary 1100 0110

   c. (2 pts) Convert 110 110 000 from binary to base 8 660

   d. (2 pts) Convert 011 111 010 from base 2 to octal 372

   e. (2 pts) Convert 25 from base 8 to binary 010 101

   f. (2 pts) Convert 1010 0000 from base 2 to decimal 160

   g. (2 pts) Convert d91c from hexadecimal to base 2 1101 1001 0001 1100

   h. (2 pts) Convert 53 from decimal to binary 0011 0101

   i. (2 pts) Convert 1011 from base 2 to base 10 11

   j. (2 pts) Convert 0111 0100 from binary to base 10 116

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  Node e;
  double f;
  int g;
  char h;
  Node *p;
  double *q;
  int *r;
  char *s;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) &h *char **

b. (2 pts) g *int*

c. (2 pts) argv[0] *char **

d. (2 pts) p->next *Node **

e. (2 pts) &s *char ***

f. (2 pts) s *char **

g. (2 pts) *r *int*

h. (2 pts) argc *int*

i. (2 pts) argv[1][2] *char*

j. (2 pts) p->next->next *Node **

k. (2 pts) p->data *int*

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt kiwi lemon
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of argv[1][0]? k

c. (2 pts) What is the value of argv[0][2]? r

d. (2 pts) What is the value of argv[2][3]? o

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 0011 0001 from base 2 to base 10 49

   b. (2 pts) Convert 1001 1000 from binary to base 10 152

   c. (2 pts) Convert 1110 1010 0010 from binary to hexadecimal ea2

   d. (2 pts) Convert 1101 1001 from base 2 to base 10 217

   e. (2 pts) Convert 101 011 001 from base 2 to octal 531

   f. (2 pts) Convert 1101 0010 1010 0001 from binary to hexadecimal d2a1

   g. (2 pts) Convert 110 110 110 from base 2 to octal 666

   h. (2 pts) Convert 158 from base 10 to binary 1001 1110

   i. (2 pts) Convert 172 from decimal to binary 1010 1100

   j. (2 pts) Convert 1010 0001 from binary to decimal 161

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double e;
  int f;
  Node g;
  char h;
  double *p;
  int *q;
  Node *r;
  char *s;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) r *Node **

b. (2 pts) *r *Node*

c. (2 pts) &p *double***

d. (2 pts) h *char*

e. (2 pts) r->next->next *Node **

f. (2 pts) argc *int*

g. (2 pts) r->next *Node **

h. (2 pts) argv[1][2] *char*

i. (2 pts) &g *Node **

j. (2 pts) argv[0] *char **

k. (2 pts) r->data *int*

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt date cherry lemon mango
```

a. (2 pts) What is the value of `argc` in this case? 5

b. (2 pts) What is the value of argv[0][3]? u

c. (2 pts) What is the value of argv[2][2]? e

d. (2 pts) What is the value of argv[1][0]? d

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 0111 0110 1000 1000 from binary to base 16 7688

   b. (2 pts) Convert 0101 0010 0110 1101 from base 2 to hexadecimal 526d

   c. (2 pts) Convert 252 from base 10 to binary 1111 1100

   d. (2 pts) Convert 0110 0101 from base 2 to decimal 101

   e. (2 pts) Convert 1001 0000 from base 2 to base 10 144

   f. (2 pts) Convert 6c7e from base 16 to base 2 0110 1100 0111 1110

   g. (2 pts) Convert 100 001 010 from binary to octal 412

   h. (2 pts) Convert 11 from octal to binary 001 001

   i. (2 pts) Convert 0111 0110 from base 2 to base 10 118

   j. (2 pts) Convert 0011 1100 from base 2 to decimal 60

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int b;
  Node c;
  double d;
  char e;
  int *f;
  Node *g;
  double *h;
  char *p;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) g->next *Node **

b. (2 pts) &c *Node **

c. (2 pts) g->next->next *Node **

d. (2 pts) g->data *int*

e. (2 pts) argv[0] *char **

f. (2 pts) c *Node*

g. (2 pts) argc *int*

h. (2 pts) *p *char*

i. (2 pts) g *Node **

j. (2 pts) argv[1][2] *char*

k. (2 pts) &h *double ***

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt apple fig grape
```

a. (2 pts) What is the value of `argc` in this case? 4

b. (2 pts) What is the value of argv[2][0]? f

c. (2 pts) What is the value of argv[0][3]? u

d. (2 pts) What is the value of argv[1][2]? p

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 1100 1110 1110 0100 from base 2 to base 16 cee4

   b. (2 pts) Convert 245 from base 10 to base 2 1111 0101

   c. (2 pts) Convert a1e4 from hexadecimal to base 2 1010 0001 1110 0100

   d. (2 pts) Convert 10 from octal to binary 001 000

   e. (2 pts) Convert 0111 1101 from base 2 to hexadecimal 7d

   f. (2 pts) Convert 0110 1101 from base 2 to base 10 109

   g. (2 pts) Convert d6ee from base 16 to binary 1101 0110 1110 1110

   h. (2 pts) Convert 1100 1100 from binary to decimal 204

   i. (2 pts) Convert 0110 1001 from binary to decimal 105

   j. (2 pts) Convert 010 010 001 from binary to base 8 221

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double s;
  Node t;
  int w;
  char x;
  double *y;
  Node *z;
  int *a;
  char *b;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) *b *char*

b. (2 pts) t *Node*

c. (2 pts) argv[1][2] *char*

d. (2 pts) z->data *int*

e. (2 pts) b *char **

f. (2 pts) argc *int*

g. (2 pts) argv[0] *char **

h. (2 pts) z->next->next *Node **

i. (2 pts) &z *Node ***

j. (2 pts) &t *Node **

k. (2 pts) z->next *Node **

3. (10 pts) In 24 hour time:

- midnight is represented by the hour 0
- 1am through 11am are represented at hour=1 through hour=11
- noon is represented by hour=12
- 1pm through 11pm are represetned by hour=13 through hour=23
- minute is a value between 0 and 59.

Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

Comply with these instructions for full credit:

- The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
  bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
  bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt banana lemon
```

a. (2 pts) What is the value of `argc` in this case? 3

b. (2 pts) What is the value of argv[1][5]? a

c. (2 pts) What is the value of argv[2][1]? e

d. (2 pts) What is the value of argv[0][4]? n

End of Exam

total points=100