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

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

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

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

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

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

   g. (2 pts) Convert 6086 from hexadecimal to base 2

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

   i. (2 pts) Convert 2fc6 from base 16 to base 2

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

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

b. (2 pts) argv[0]

c. (2 pts) *e

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

e. (2 pts) z

f. (2 pts) b->next

g. (2 pts) argc

h. (2 pts) b->data

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

j. (2 pts) d

k. (2 pts) &a

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);
```

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

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

Comply with these instructions for full credit:

- 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?

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

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

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

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

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

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

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

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

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

   g. (2 pts) Convert 62b4 from base 16 to binary

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

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

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

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]

b. (2 pts) argv[0]

c. (2 pts) q

d. (2 pts) *t

e. (2 pts) &h

f. (2 pts) argc

g. (2 pts) t->data

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

i. (2 pts) t

j. (2 pts) t->next

k. (2 pts) &t

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?

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

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

d. (2 pts) What is the value of argv[2][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 1111 1010 0101 0110 from base 2 to hexadecimal

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

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

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

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

   f. (2 pts) Convert 2670 from hexadecimal to base 2

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

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

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

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

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

b. (2 pts) p

c. (2 pts) *h

d. (2 pts) &e

e. (2 pts) argc

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

g. (2 pts) argv[0]

h. (2 pts) e

i. (2 pts) h->data

j. (2 pts) h->next

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

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?

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

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

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

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
   b. (2 pts) Convert 111 011 001 from base 2 to octal
   c. (2 pts) Convert 0001 0010 from base 2 to decimal
   d. (2 pts) Convert 152 from base 10 to binary
   e. (2 pts) Convert 100 011 110 from binary to octal
   f. (2 pts) Convert 001 001 111 from binary to base 8
   g. (2 pts) Convert 0101 1110 0101 1000 from base 2 to hexadecimal
   h. (2 pts) Convert 6 from base 8 to base 2
   i. (2 pts) Convert 8e0c from base 16 to binary
   j. (2 pts) Convert 101 111 111 from binary to octal

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]

b. (2 pts) &w

c. (2 pts) a->data

d. (2 pts) argc

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

f. (2 pts) a->next

g. (2 pts) w

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

i. (2 pts) &d

j. (2 pts) *d

k. (2 pts) d

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?

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

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

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

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
   b. (2 pts) Convert 1010 0110 from binary to decimal
   c. (2 pts) Convert 0000 from binary to decimal
   d. (2 pts) Convert 36 from base 10 to base 2
   e. (2 pts) Convert 114 from decimal to binary
   f. (2 pts) Convert 110 000 011 from binary to octal
   g. (2 pts) Convert 010 000 from binary to octal
   h. (2 pts) Convert 160 from decimal to base 2
   i. (2 pts) Convert 88 from decimal to base 2
   j. (2 pts) Convert 010 110 111 from binary to octal

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

b. (2 pts) y->next

c. (2 pts) argc

d. (2 pts) argv[0]

e. (2 pts) z

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

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

h. (2 pts) y->data

i. (2 pts) &w

j. (2 pts) &z

k. (2 pts) *y

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?

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

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

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

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

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

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

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

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

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

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

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

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

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

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

c. (2 pts) z->data

d. (2 pts) *x

e. (2 pts) &a

f. (2 pts) &t

g. (2 pts) y

h. (2 pts) z->next

i. (2 pts) argv[0]

j. (2 pts) r

k. (2 pts) argc

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?

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

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

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

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

   b. (2 pts) Convert 328e from base 16 to base 2

   c. (2 pts) Convert 250b from base 16 to binary

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

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

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

   g. (2 pts) Convert b467 from hexadecimal to binary

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

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

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

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

b. (2 pts) d

c. (2 pts) b

d. (2 pts) *f

e. (2 pts) argv[0]

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

g. (2 pts) &e

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

i. (2 pts) argc

j. (2 pts) &a

k. (2 pts) f->data

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?

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

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

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

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

   b. (2 pts) Convert 9041 from base 16 to binary

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

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

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

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

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

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

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

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

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]

b. (2 pts) &e

c. (2 pts) &y

d. (2 pts) *d

e. (2 pts) c->data

f. (2 pts) c->next

g. (2 pts) y

h. (2 pts) argc

i. (2 pts) d

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

k. (2 pts) argv[0]

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?

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

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

d. (2 pts) What is the value of argv[1][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 3 from octal to base 2

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

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

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

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

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

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

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

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

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

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

b. (2 pts) &z

c. (2 pts) argc

d. (2 pts) z->data

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

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

g. (2 pts) *b

h. (2 pts) &s

i. (2 pts) z->next

j. (2 pts) argv[0]

k. (2 pts) x

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?

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

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

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
   b. (2 pts) Convert 110 111 from binary to octal
   c. (2 pts) Convert 32a0 from hexadecimal to binary
   d. (2 pts) Convert 001 100 110 from binary to octal
   e. (2 pts) Convert 111 011 001 from base 2 to base 8
   f. (2 pts) Convert 111 111 from base 2 to base 8
   g. (2 pts) Convert 0111 0100 0010 0101 from binary to hexadecimal
   h. (2 pts) Convert 3246 from base 16 to binary
   i. (2 pts) Convert 67 from base 8 to base 2
   j. (2 pts) Convert 0111 1010 from binary to decimal

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

b. (2 pts) argc

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

d. (2 pts) q->next

e. (2 pts) &g

f. (2 pts) argv[0]

g. (2 pts) e

h. (2 pts) *s

i. (2 pts) q->data

j. (2 pts) &r

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

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?

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

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

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

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
   b. (2 pts) Convert 110 101 100 from base 2 to base 8
   c. (2 pts) Convert 2081 from hexadecimal to binary
   d. (2 pts) Convert 101 111 110 from binary to base 8
   e. (2 pts) Convert 207 from base 10 to binary
   f. (2 pts) Convert 56 from base 8 to base 2
   g. (2 pts) Convert 0001 1110 0001 0110 from base 2 to hexadecimal
   h. (2 pts) Convert 56 from octal to binary
   i. (2 pts) Convert 52 from octal to binary
   j. (2 pts) Convert 0001 0110 from base 2 to decimal

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]

b. (2 pts) g->data

c. (2 pts) argv[0]

d. (2 pts) &f

e. (2 pts) g->next

f. (2 pts) c

g. (2 pts) *e

h. (2 pts) h

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

j. (2 pts) argc

k. (2 pts) &b

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);
```

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

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

Comply with these instructions for full credit:

- 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?

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

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

d. (2 pts) What is the value of argv[2][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 110 111 011 from base 2 to base 8

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

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

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

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

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

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

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

   i. (2 pts) Convert 9be1 from base 16 to base 2

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

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

b. (2 pts) &s

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

d. (2 pts) y

e. (2 pts) &a

f. (2 pts) x->next

g. (2 pts) argv[0]

h. (2 pts) t

i. (2 pts) *x

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

k. (2 pts) x->data

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?

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

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

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

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
   b. (2 pts) Convert 51 from base 10 to binary
   c. (2 pts) Convert 179 from decimal to binary
   d. (2 pts) Convert 6 from base 10 to binary
   e. (2 pts) Convert 35 from base 10 to base 2
   f. (2 pts) Convert 84 from decimal to binary
   g. (2 pts) Convert 110 011 from binary to octal
   h. (2 pts) Convert e819 from hexadecimal to binary
   i. (2 pts) Convert 0110 0101 1111 1000 from binary to hexadecimal
   j. (2 pts) Convert 101 111 011 from base 2 to base 8

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

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

c. (2 pts) h

d. (2 pts) s->next

e. (2 pts) argv[0]

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

g. (2 pts) s->data

h. (2 pts) *t

i. (2 pts) argc

j. (2 pts) s

k. (2 pts) &h

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?

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

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

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

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
   b. (2 pts) Convert 0100 from binary to decimal
   c. (2 pts) Convert 111 010 100 from binary to base 8
   d. (2 pts) Convert 630d from hexadecimal to binary
   e. (2 pts) Convert 79ba from base 16 to binary
   f. (2 pts) Convert 8751 from base 16 to binary
   g. (2 pts) Convert 110 111 from binary to base 8
   h. (2 pts) Convert 24 from base 8 to binary
   i. (2 pts) Convert 0111 1011 0011 from base 2 to base 16
   j. (2 pts) Convert 1110 1010 from base 2 to decimal

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

b. (2 pts) argc

c. (2 pts) *f

d. (2 pts) b

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

f. (2 pts) &d

g. (2 pts) e->data

h. (2 pts) argv[0]

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

j. (2 pts) e->next

k. (2 pts) &h

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);
```

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

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

Comply with these instructions for full credit:

- 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?

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

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

d. (2 pts) What is the value of argv[2][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 98e4 from base 16 to binary

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

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

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

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

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

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

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

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

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

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

b. (2 pts) &z

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

d. (2 pts) a

e. (2 pts) d->next

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

g. (2 pts) *d

h. (2 pts) &c

i. (2 pts) argc

j. (2 pts) d->data

k. (2 pts) argv[0]

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?

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

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

d. (2 pts) What is the value of argv[2][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 1111 0001 0100 0001 from binary to hexadecimal

   b. (2 pts) Convert 011 000 011 from binary to base 8

   c. (2 pts) Convert 37 from octal to binary

   d. (2 pts) Convert 1010 1010 from base 2 to base 10

   e. (2 pts) Convert 1100 1101 from base 2 to base 10

   f. (2 pts) Convert 001 000 100 from base 2 to base 8

   g. (2 pts) Convert 0001 0111 from base 2 to base 10

   h. (2 pts) Convert 127 from base 10 to binary

   i. (2 pts) Convert 61 from base 8 to base 2

   j. (2 pts) Convert 44 from base 8 to binary

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

b. (2 pts) s->data

c. (2 pts) argv[0]

d. (2 pts) argv[1][2]

e. (2 pts) *w

f. (2 pts) s->next

g. (2 pts) argc

h. (2 pts) &q

i. (2 pts) s->next->next

j. (2 pts) &s

k. (2 pts) r

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);
```

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

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

Comply with these instructions for full credit:

- 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?

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

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

d. (2 pts) What is the value of argv[1][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 001 101 100 from base 2 to base 8
   b. (2 pts) Convert 1bb0 from base 16 to binary
   c. (2 pts) Convert 6b44 from base 16 to binary
   d. (2 pts) Convert 15 from base 8 to binary
   e. (2 pts) Convert 176 from base 10 to binary
   f. (2 pts) Convert 57 from octal to base 2
   g. (2 pts) Convert c17c from hexadecimal to base 2
   h. (2 pts) Convert 1 from octal to binary
   i. (2 pts) Convert 8e55 from hexadecimal to binary
   j. (2 pts) Convert 13 from octal to binary

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

b. (2 pts) e

c. (2 pts) p->data

d. (2 pts) p->next->next

e. (2 pts) &e

f. (2 pts) argv[0]

g. (2 pts) h

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

i. (2 pts) p->next

j. (2 pts) argc

k. (2 pts) *r

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?

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

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

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
   b. (2 pts) Convert ed35 from hexadecimal to base 2
   c. (2 pts) Convert 101 000 011 from base 2 to octal
   d. (2 pts) Convert 146 from base 10 to base 2
   e. (2 pts) Convert 6e7 from base 16 to binary
   f. (2 pts) Convert 1110 1110 from binary to decimal
   g. (2 pts) Convert 1100 0011 from base 2 to decimal
   h. (2 pts) Convert 33 from base 8 to binary
   i. (2 pts) Convert 300f from base 16 to base 2
   j. (2 pts) Convert 010 110 011 from binary to base 8

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

b. (2 pts) g

c. (2 pts) r->next

d. (2 pts) *s

e. (2 pts) argv[1][2]

f. (2 pts) r->next->next

g. (2 pts) argv[0]

h. (2 pts) s

i. (2 pts) &p

j. (2 pts) &h

k. (2 pts) argc

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?

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

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

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

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

   b. (2 pts) Convert 51 from octal to binary

   c. (2 pts) Convert 1000 1111 from base 2 to decimal

   d. (2 pts) Convert 0001 1110 from base 2 to decimal

   e. (2 pts) Convert 72 from base 8 to base 2

   f. (2 pts) Convert 100 010 001 from base 2 to octal

   g. (2 pts) Convert 0110 1101 from binary to decimal

   h. (2 pts) Convert 111 101 110 from base 2 to octal

   i. (2 pts) Convert 1111 1010 0010 0110 from binary to hexadecimal

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

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]

b. (2 pts) &z

c. (2 pts) argc

d. (2 pts) y->next->next

e. (2 pts) y->data

f. (2 pts) s

g. (2 pts) z

h. (2 pts) argv[0]

i. (2 pts) &t

j. (2 pts) *x

k. (2 pts) y->next

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:

- 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.

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

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

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?

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

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

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

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

   b. (2 pts) Convert 0100 1010 0010 1010 from binary to base 16

   c. (2 pts) Convert 52 from base 10 to base 2

   d. (2 pts) Convert 1101 1010 0011 0011 from base 2 to hexadecimal

   e. (2 pts) Convert 0101 1010 from base 2 to decimal

   f. (2 pts) Convert 1000 1001 1111 0010 from base 2 to base 16

   g. (2 pts) Convert bf4e from base 16 to binary

   h. (2 pts) Convert 157 from base 10 to base 2

   i. (2 pts) Convert 1110 1100 1001 1010 from binary to hexadecimal

   j. (2 pts) Convert 0 from octal to binary

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

b. (2 pts) p->data

c. (2 pts) p

d. (2 pts) argv[0]

e. (2 pts) d

f. (2 pts) &p

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

h. (2 pts) argc

i. (2 pts) *h

j. (2 pts) p->next

k. (2 pts) &f

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?

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

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

d. (2 pts) What is the value of argv[1][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 0100 1001 1001 1101 from binary to hexadecimal
   b. (2 pts) Convert 1 from base 8 to binary
   c. (2 pts) Convert 001 000 101 from base 2 to base 8
   d. (2 pts) Convert 31 from base 8 to base 2
   e. (2 pts) Convert 3df7 from base 16 to binary
   f. (2 pts) Convert 23cf from base 16 to base 2
   g. (2 pts) Convert 0110 1001 0011 1111 from binary to hexadecimal
   h. (2 pts) Convert 11 from base 8 to base 2
   i. (2 pts) Convert b6b1 from base 16 to base 2
   j. (2 pts) Convert 47 from base 8 to base 2

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]

b. (2 pts) *f

c. (2 pts) e->next

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

e. (2 pts) argv[0]

f. (2 pts) &h

g. (2 pts) c

h. (2 pts) e

i. (2 pts) &b

j. (2 pts) e->data

k. (2 pts) argc

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);
```

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

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

Comply with these instructions for full credit:

- 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?

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

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

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

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

   b. (2 pts) Convert 1101 0101 1101 1000 from base 2 to base 16

   c. (2 pts) Convert 5925 from base 16 to base 2

   d. (2 pts) Convert 110 000 100 from base 2 to octal

   e. (2 pts) Convert 1001 0100 0001 0100 from base 2 to hexadecimal

   f. (2 pts) Convert 0101 0110 0011 1000 from binary to hexadecimal

   g. (2 pts) Convert 6b6d from base 16 to base 2

   h. (2 pts) Convert 43 from octal to base 2

   i. (2 pts) Convert 0101 1000 0110 1100 from binary to base 16

   j. (2 pts) Convert 62 from base 8 to binary

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

b. (2 pts) d

c. (2 pts) argv[0]

d. (2 pts) g->next

e. (2 pts) &a

f. (2 pts) h

g. (2 pts) argc

h. (2 pts) *f

i. (2 pts) g->data

j. (2 pts) g->next->next

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

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?

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

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

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

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
   b. (2 pts) Convert 1001 0000 from binary to decimal
   c. (2 pts) Convert 0100 0111 from binary to base 10
   d. (2 pts) Convert 010 011 100 from base 2 to base 8
   e. (2 pts) Convert 011 101 110 from base 2 to base 8
   f. (2 pts) Convert 74 from base 8 to binary
   g. (2 pts) Convert 155d from hexadecimal to base 2
   h. (2 pts) Convert 22 from base 10 to base 2
   i. (2 pts) Convert 2283 from base 16 to base 2
   j. (2 pts) Convert 31 from base 8 to binary

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

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

c. (2 pts) b->data

d. (2 pts) &e

e. (2 pts) b->next

f. (2 pts) &a

g. (2 pts) d

h. (2 pts) z

i. (2 pts) argc

j. (2 pts) argv[0]

k. (2 pts) b->next->next

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?

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

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

d. (2 pts) What is the value of argv[2][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 5207 from hexadecimal to binary
   b. (2 pts) Convert 1111 0010 0100 0000 from binary to hexadecimal
   c. (2 pts) Convert 0100 1001 from base 2 to decimal
   d. (2 pts) Convert 45b8 from hexadecimal to binary
   e. (2 pts) Convert 3358 from hexadecimal to binary
   f. (2 pts) Convert c3b5 from hexadecimal to binary
   g. (2 pts) Convert 0111 0110 from binary to decimal
   h. (2 pts) Convert c3ce from base 16 to base 2
   i. (2 pts) Convert 9d33 from hexadecimal to base 2
   j. (2 pts) Convert 1010 1101 0110 1111 from binary to base 16

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

b. (2 pts) &s

c. (2 pts) *r

d. (2 pts) p->data

e. (2 pts) s

f. (2 pts) p->next->next

g. (2 pts) &g

h. (2 pts) argv[0]

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

j. (2 pts) p->next

k. (2 pts) g

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);
```

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

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

Comply with these instructions for full credit:

- 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?

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

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

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

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

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

   c. (2 pts) Convert 55 from base 10 to binary

   d. (2 pts) Convert d1a2 from hexadecimal to binary

   e. (2 pts) Convert 5 from octal to base 2

   f. (2 pts) Convert 5d92 from base 16 to binary

   g. (2 pts) Convert 0010 0000 from base 2 to decimal

   h. (2 pts) Convert 22 from base 8 to base 2

   i. (2 pts) Convert 674a from hexadecimal to binary

   j. (2 pts) Convert 0100 1001 0101 1001 from base 2 to base 16

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]

b. (2 pts) &g

c. (2 pts) g->next->next

d. (2 pts) argv[1][2]

e. (2 pts) g->next

f. (2 pts) g

g. (2 pts) g->data

h. (2 pts) &b

i. (2 pts) argc

j. (2 pts) a

k. (2 pts) *e

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?

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

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

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

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
   b. (2 pts) Convert 011 111 011 from base 2 to octal
   c. (2 pts) Convert 6dd5 from base 16 to base 2
   d. (2 pts) Convert 0010 1101 1110 0100 from base 2 to hexadecimal
   e. (2 pts) Convert 011 011 001 from binary to base 8
   f. (2 pts) Convert 43 from octal to base 2
   g. (2 pts) Convert 0010 0011 0001 1110 from binary to base 16
   h. (2 pts) Convert 55 from octal to base 2
   i. (2 pts) Convert 905 from hexadecimal to binary
   j. (2 pts) Convert 35 from base 8 to base 2

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

b. (2 pts) *p

c. (2 pts) c

d. (2 pts) g

e. (2 pts) g->data

f. (2 pts) argv[0]

g. (2 pts) &p

h. (2 pts) g->next->next

i. (2 pts) g->next

j. (2 pts) argv[1][2]

k. (2 pts) argc

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?

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

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

d. (2 pts) What is the value of argv[1][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 1101 from binary to base 10
   b. (2 pts) Convert 3818 from hexadecimal to base 2
   c. (2 pts) Convert 5bb6 from hexadecimal to binary
   d. (2 pts) Convert 1011 1001 1100 1110 from binary to base 16
   e. (2 pts) Convert 80 from decimal to base 2
   f. (2 pts) Convert 0010 1001 1101 1000 from binary to hexadecimal
   g. (2 pts) Convert 63 from base 8 to binary
   h. (2 pts) Convert 001 111 000 from binary to base 8
   i. (2 pts) Convert 1101 0011 0001 1100 from base 2 to hexadecimal
   j. (2 pts) Convert 0001 0001 1000 0100 from binary to base 16

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]

b. (2 pts) &b

c. (2 pts) *d

d. (2 pts) d->data

e. (2 pts) &d

f. (2 pts) argc

g. (2 pts) y

h. (2 pts) c

i. (2 pts) d->next->next

j. (2 pts) argv[1][2]

k. (2 pts) d->next

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);
```

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

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

Comply with these instructions for full credit:

- 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?

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

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

d. (2 pts) What is the value of argv[1][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 0110 0101 from binary to base 10
   b. (2 pts) Convert 66 from octal to base 2
   c. (2 pts) Convert 0001 0001 1000 from base 2 to base 16
   d. (2 pts) Convert 117 from decimal to base 2
   e. (2 pts) Convert 1100 0000 1000 0101 from binary to hexadecimal
   f. (2 pts) Convert 001 010 110 from binary to octal
   g. (2 pts) Convert 0001 1110 from binary to base 10
   h. (2 pts) Convert e28f from hexadecimal to binary
   i. (2 pts) Convert 1100 0101 1000 1111 from base 2 to base 16
   j. (2 pts) Convert 29 from base 10 to binary

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]

b. (2 pts) argv[0]

c. (2 pts) t->next

d. (2 pts) argc

e. (2 pts) *t

f. (2 pts) t->data

g. (2 pts) t->next->next

h. (2 pts) s

i. (2 pts) y

j. (2 pts) &t

k. (2 pts) &r

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?

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

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

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
   b. (2 pts) Convert 149 from decimal to binary
   c. (2 pts) Convert 238 from decimal to base 2
   d. (2 pts) Convert 010 from binary to octal
   e. (2 pts) Convert 101 000 111 from binary to base 8
   f. (2 pts) Convert 1100 0101 from base 2 to decimal
   g. (2 pts) Convert 1100 1000 from base 2 to base 10
   h. (2 pts) Convert 32 from base 8 to binary
   i. (2 pts) Convert 8fa6 from hexadecimal to binary
   j. (2 pts) Convert 1011 1001 from binary to decimal

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

b. (2 pts) argv[0]

c. (2 pts) d

d. (2 pts) argv[1][2]

e. (2 pts) d->data

f. (2 pts) &d

g. (2 pts) *d

h. (2 pts) argc

i. (2 pts) y

j. (2 pts) &z

k. (2 pts) d->next

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?

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

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

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

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

   b. (2 pts) Convert 0110 0110 from base 2 to base 10

   c. (2 pts) Convert 001 001 010 from binary to octal

   d. (2 pts) Convert 0101 1101 from binary to base 10

   e. (2 pts) Convert 1111 1001 from binary to base 10

   f. (2 pts) Convert 247 from base 10 to base 2

   g. (2 pts) Convert 1100 1010 from binary to decimal

   h. (2 pts) Convert 110 100 110 from binary to base 8

   i. (2 pts) Convert 49 from base 10 to base 2

   j. (2 pts) Convert 229 from decimal to base 2

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

b. (2 pts) &b

c. (2 pts) b->next

d. (2 pts) argv[0]

e. (2 pts) argc

f. (2 pts) c

g. (2 pts) &x

h. (2 pts) b->next->next

i. (2 pts) *c

j. (2 pts) a

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

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?

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

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

d. (2 pts) What is the value of argv[1][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 10 from base 8 to base 2
   b. (2 pts) Convert 0010 0000 1011 1011 from binary to hexadecimal
   c. (2 pts) Convert 0001 0011 0011 0111 from base 2 to hexadecimal
   d. (2 pts) Convert 233 from decimal to binary
   e. (2 pts) Convert 221 from decimal to binary
   f. (2 pts) Convert 1001 0001 from binary to base 10
   g. (2 pts) Convert 0111 0100 from base 2 to base 10
   h. (2 pts) Convert 90 from decimal to binary
   i. (2 pts) Convert fb78 from hexadecimal to binary
   j. (2 pts) Convert 129 from base 10 to binary

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

b. (2 pts) y->next

c. (2 pts) y->data

d. (2 pts) argc

e. (2 pts) argv[1][2]

f. (2 pts) &x

g. (2 pts) argv[0]

h. (2 pts) *b

i. (2 pts) &a

j. (2 pts) y

k. (2 pts) y->next->next

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?

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

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

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

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
   b. (2 pts) Convert c387 from hexadecimal to base 2
   c. (2 pts) Convert 56 from octal to base 2
   d. (2 pts) Convert 51 from octal to binary
   e. (2 pts) Convert 0100 1101 from binary to base 10
   f. (2 pts) Convert 44 from octal to base 2
   g. (2 pts) Convert 1100 0110 1000 0100 from binary to hexadecimal
   h. (2 pts) Convert 151 from hexadecimal to binary
   i. (2 pts) Convert edec from base 16 to base 2
   j. (2 pts) Convert 1000 1100 from binary to base 10

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]

b. (2 pts) q->next->next

c. (2 pts) &h

d. (2 pts) *r

e. (2 pts) e

f. (2 pts) q->next

g. (2 pts) s

h. (2 pts) &q

i. (2 pts) q->data

j. (2 pts) argc

k. (2 pts) argv[0]

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);
```

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

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

Comply with these instructions for full credit:

- 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?

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

c. (2 pts) What is the value of argv[1][5]?

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

   b. (2 pts) Convert 0111 1101 1010 1111 from binary to hexadecimal

   c. (2 pts) Convert 51 from base 8 to binary

   d. (2 pts) Convert 0011 0000 1111 0100 from binary to base 16

   e. (2 pts) Convert 30fe from hexadecimal to binary

   f. (2 pts) Convert 0010 1100 0111 1001 from binary to hexadecimal

   g. (2 pts) Convert 0111 0000 0111 0101 from base 2 to hexadecimal

   h. (2 pts) Convert 100 010 010 from binary to octal

   i. (2 pts) Convert 184 from decimal to binary

   j. (2 pts) Convert 0010 1000 from base 2 to base 10

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]

b. (2 pts) g

c. (2 pts) &d

d. (2 pts) *f

e. (2 pts) h->next

f. (2 pts) e

g. (2 pts) &p

h. (2 pts) h->next->next

i. (2 pts) argc

j. (2 pts) argv[1][2]

k. (2 pts) h->data

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?

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

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

d. (2 pts) What is the value of argv[1][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 111 011 110 from binary to octal

   b. (2 pts) Convert 23 from base 8 to base 2

   c. (2 pts) Convert 1101 1100 from base 2 to base 10

   d. (2 pts) Convert 1000 1101 from binary to base 10

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

   f. (2 pts) Convert 010 111 101 from base 2 to base 8

   g. (2 pts) Convert 72a3 from hexadecimal to binary

   h. (2 pts) Convert 111 100 011 from base 2 to octal

   i. (2 pts) Convert 0101 1001 1011 1101 from base 2 to base 16

   j. (2 pts) Convert 0101 0100 from base 2 to decimal

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

b. (2 pts) argv[0]

c. (2 pts) argc

d. (2 pts) &y

e. (2 pts) z->next

f. (2 pts) z->data

g. (2 pts) z

h. (2 pts) &t

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

j. (2 pts) *b

k. (2 pts) s

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:

- 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.

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

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

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?

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

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

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

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
   b. (2 pts) Convert 010 010 from base 2 to octal
   c. (2 pts) Convert 1100 1010 from base 2 to base 10
   d. (2 pts) Convert 6 from octal to binary
   e. (2 pts) Convert 6a68 from hexadecimal to binary
   f. (2 pts) Convert 111 110 001 from binary to base 8
   g. (2 pts) Convert 0001 1100 1001 0100 from base 2 to hexadecimal
   h. (2 pts) Convert 121 from base 10 to binary
   i. (2 pts) Convert 35 from decimal to binary
   j. (2 pts) Convert 1111 0000 from base 2 to decimal

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

b. (2 pts) s

c. (2 pts) &h

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

e. (2 pts) q

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

g. (2 pts) *t

h. (2 pts) s->next

i. (2 pts) s->data

j. (2 pts) argv[0]

k. (2 pts) &x

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);
```

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

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

Comply with these instructions for full credit:

- 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?

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

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

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

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

   b. (2 pts) Convert 1010 1100 0010 1010 from binary to base 16

   c. (2 pts) Convert 112 from base 10 to binary

   d. (2 pts) Convert d4b3 from base 16 to binary

   e. (2 pts) Convert 218 from decimal to base 2

   f. (2 pts) Convert 1111 1010 from binary to base 10

   g. (2 pts) Convert 0110 1110 0100 0111 from binary to base 16

   h. (2 pts) Convert 10 from base 8 to binary

   i. (2 pts) Convert 22 from base 10 to base 2

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

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]

b. (2 pts) &f

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

d. (2 pts) *e

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

f. (2 pts) &d

g. (2 pts) e->data

h. (2 pts) argc

i. (2 pts) g

j. (2 pts) b

k. (2 pts) e->next

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?

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

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

d. (2 pts) What is the value of argv[2][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 64 from base 8 to base 2
   b. (2 pts) Convert 0110 0110 from binary to base 10
   c. (2 pts) Convert 010 111 011 from base 2 to octal
   d. (2 pts) Convert 609d from base 16 to base 2
   e. (2 pts) Convert be2c from hexadecimal to binary
   f. (2 pts) Convert 147 from decimal to base 2
   g. (2 pts) Convert 110 000 from base 2 to base 8
   h. (2 pts) Convert 101 001 111 from binary to octal
   i. (2 pts) Convert e05f from hexadecimal to binary
   j. (2 pts) Convert 100 110 000 from binary to octal

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

b. (2 pts) argv[0]

c. (2 pts) d->next

d. (2 pts) *d

e. (2 pts) d->data

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

g. (2 pts) &b

h. (2 pts) &a

i. (2 pts) d->next->next

j. (2 pts) c

k. (2 pts) x

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?

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

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

d. (2 pts) What is the value of argv[2][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 1100 0000 from binary to hexadecimal

   b. (2 pts) Convert 001 101 111 from base 2 to base 8

   c. (2 pts) Convert 945a from base 16 to binary

   d. (2 pts) Convert 1011 1100 1101 1110 from base 2 to hexadecimal

   e. (2 pts) Convert 0001 0100 0100 1010 from binary to hexadecimal

   f. (2 pts) Convert c656 from base 16 to binary

   g. (2 pts) Convert 0001 1010 0110 0110 from binary to base 16

   h. (2 pts) Convert 16 from base 10 to binary

   i. (2 pts) Convert 821a from base 16 to base 2

   j. (2 pts) Convert 110 001 001 from binary to base 8

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

b. (2 pts) argv[0]

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

d. (2 pts) argc

e. (2 pts) &t

f. (2 pts) p

g. (2 pts) s

h. (2 pts) w->next

i. (2 pts) *t

j. (2 pts) w->data

k. (2 pts) &p

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);
```

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

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

Comply with these instructions for full credit:

- 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?

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

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

d. (2 pts) What is the value of argv[1][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 010 001 111 from binary to octal

   b. (2 pts) Convert 1111 0010 from base 2 to decimal

   c. (2 pts) Convert 130 from decimal to binary

   d. (2 pts) Convert 0100 1000 1100 1000 from base 2 to base 16

   e. (2 pts) Convert 111 101 111 from binary to base 8

   f. (2 pts) Convert 96 from base 10 to base 2

   g. (2 pts) Convert 1100 0100 0101 0110 from base 2 to base 16

   h. (2 pts) Convert 9862 from base 16 to binary

   i. (2 pts) Convert 76 from decimal to binary

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

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

b. (2 pts) *s

c. (2 pts) q->data

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

e. (2 pts) argv[0]

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

g. (2 pts) &s

h. (2 pts) s

i. (2 pts) &e

j. (2 pts) q->next

k. (2 pts) argc

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?

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

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

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

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

   b. (2 pts) Convert 79 from base 10 to base 2

   c. (2 pts) Convert 011 010 111 from binary to base 8

   d. (2 pts) Convert 41 from octal to binary

   e. (2 pts) Convert 77 from octal to binary

   f. (2 pts) Convert 0010 0101 from base 2 to base 10

   g. (2 pts) Convert 65 from base 8 to binary

   h. (2 pts) Convert 010 010 001 from base 2 to base 8

   i. (2 pts) Convert 1111 1011 1100 0001 from base 2 to hexadecimal

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

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

b. (2 pts) h->next->next

c. (2 pts) h->data

d. (2 pts) h

e. (2 pts) argc

f. (2 pts) argv[0]

g. (2 pts) d

h. (2 pts) *q

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

j. (2 pts) &h

k. (2 pts) &c

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?

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

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

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

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

   b. (2 pts) Convert 2 from octal to binary

   c. (2 pts) Convert 010 110 010 from base 2 to base 8

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

   e. (2 pts) Convert 111 000 011 from binary to base 8

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

   g. (2 pts) Convert 40 from octal to binary

   h. (2 pts) Convert 110 100 000 from base 2 to base 8

   i. (2 pts) Convert 1100 0101 1101 1000 from base 2 to base 16

   j. (2 pts) Convert 1000 1010 from base 2 to decimal

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

b. (2 pts) &x

c. (2 pts) s

d. (2 pts) w->next

e. (2 pts) w->next->next

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

g. (2 pts) &r

h. (2 pts) h

i. (2 pts) argv[0]

j. (2 pts) *s

k. (2 pts) argc

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?

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

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

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

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

   b. (2 pts) Convert 1101 1011 0110 0010 from binary to base 16

   c. (2 pts) Convert 1000 1111 1100 1111 from base 2 to hexadecimal

   d. (2 pts) Convert 0110 1111 from binary to decimal

   e. (2 pts) Convert 0011 1000 from binary to base 10

   f. (2 pts) Convert f22d from base 16 to base 2

   g. (2 pts) Convert 1000 0100 0001 0100 from base 2 to base 16

   h. (2 pts) Convert 3967 from hexadecimal to base 2

   i. (2 pts) Convert 0110 0111 1001 0011 from base 2 to hexadecimal

   j. (2 pts) Convert 182 from base 10 to binary

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

b. (2 pts) argc

c. (2 pts) &t

d. (2 pts) w->next

e. (2 pts) argv[1][2]

f. (2 pts) *t

g. (2 pts) q

h. (2 pts) &r

i. (2 pts) w->data

j. (2 pts) y

k. (2 pts) argv[0]

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);
```

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

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

Comply with these instructions for full credit:

- 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?

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

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

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

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
   b. (2 pts) Convert 45 from base 8 to binary
   c. (2 pts) Convert 125 from decimal to binary
   d. (2 pts) Convert 76 from octal to base 2
   e. (2 pts) Convert 1b57 from hexadecimal to base 2
   f. (2 pts) Convert 140 from base 10 to base 2
   g. (2 pts) Convert 0010 1110 0000 0101 from binary to base 16
   h. (2 pts) Convert c118 from hexadecimal to base 2
   i. (2 pts) Convert 14 from octal to base 2
   j. (2 pts) Convert 0101 0010 from binary to decimal

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

b. (2 pts) argv[0]

c. (2 pts) &t

d. (2 pts) q->data

e. (2 pts) &g

f. (2 pts) argc

g. (2 pts) *r

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

i. (2 pts) p

j. (2 pts) s

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

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?

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

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

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

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

   b. (2 pts) Convert 3857 from base 16 to base 2

   c. (2 pts) Convert 2312 from base 16 to base 2

   d. (2 pts) Convert b6be from base 16 to base 2

   e. (2 pts) Convert 42 from octal to base 2

   f. (2 pts) Convert 1000 1101 0101 1010 from base 2 to base 16

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

   h. (2 pts) Convert 011 001 111 from binary to base 8

   i. (2 pts) Convert 11 from octal to binary

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

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

b. (2 pts) c

c. (2 pts) c->next->next

d. (2 pts) argc

e. (2 pts) &d

f. (2 pts) c->next

g. (2 pts) c->data

h. (2 pts) argv[0]

i. (2 pts) &y

j. (2 pts) argv[1][2]

k. (2 pts) *d

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);
```

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

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

Comply with these instructions for full credit:

- 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?

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

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

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

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
   b. (2 pts) Convert 111 100 101 from base 2 to base 8
   c. (2 pts) Convert 0001 0000 from binary to decimal
   d. (2 pts) Convert 42a8 from base 16 to binary
   e. (2 pts) Convert 0110 1111 from binary to base 10
   f. (2 pts) Convert 11 from octal to base 2
   g. (2 pts) Convert 0010 1001 from binary to base 10
   h. (2 pts) Convert 239 from base 10 to base 2
   i. (2 pts) Convert 1110 1110 0011 0101 from base 2 to base 16
   j. (2 pts) Convert 111 110 100 from binary to base 8

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

b. (2 pts) *b

c. (2 pts) b->next

d. (2 pts) argc

e. (2 pts) argv[0]

f. (2 pts) b->data

g. (2 pts) c

h. (2 pts) &z

i. (2 pts) &t

j. (2 pts) b->next->next

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

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?

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

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

d. (2 pts) What is the value of argv[2][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 43 from base 8 to binary

   b. (2 pts) Convert 61 from octal to base 2

   c. (2 pts) Convert 21 from octal to base 2

   d. (2 pts) Convert 1001 1110 1110 1001 from binary to hexadecimal

   e. (2 pts) Convert 197 from decimal to base 2

   f. (2 pts) Convert 26 from octal to binary

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

   h. (2 pts) Convert 5829 from base 16 to base 2

   i. (2 pts) Convert 43 from octal to binary

   j. (2 pts) Convert 0010 0110 from binary to decimal

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

b. (2 pts) p->data

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

d. (2 pts) &g

e. (2 pts) &q

f. (2 pts) s

g. (2 pts) argv[0]

h. (2 pts) p->next->next

i. (2 pts) argc

j. (2 pts) *r

k. (2 pts) p->next

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?

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

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

d. (2 pts) What is the value of argv[1][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 1101 0010 from binary to decimal
   b. (2 pts) Convert 126 from base 10 to binary
   c. (2 pts) Convert 3531 from hexadecimal to base 2
   d. (2 pts) Convert 0010 1010 1101 0100 from binary to hexadecimal
   e. (2 pts) Convert 1010 1000 1000 0100 from binary to hexadecimal
   f. (2 pts) Convert 74 from base 8 to base 2
   g. (2 pts) Convert 1101 0101 from binary to decimal
   h. (2 pts) Convert 67 from octal to base 2
   i. (2 pts) Convert 0101 1010 0000 0110 from base 2 to base 16
   j. (2 pts) Convert 1100 0010 from base 2 to base 10

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

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

c. (2 pts) *p

d. (2 pts) f->data

e. (2 pts) f->next

f. (2 pts) argv[0]

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

h. (2 pts) &c

i. (2 pts) &p

j. (2 pts) argc

k. (2 pts) g

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);
```

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

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

Comply with these instructions for full credit:

- 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?

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

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

d. (2 pts) What is the value of argv[1][4]?

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

   b. (2 pts) Convert 0010 0000 from base 2 to decimal

   c. (2 pts) Convert 1101 1010 1001 0011 from binary to hexadecimal

   d. (2 pts) Convert 230 from base 10 to base 2

   e. (2 pts) Convert 110 001 from binary to base 8

   f. (2 pts) Convert 244 from base 10 to base 2

   g. (2 pts) Convert 39 from base 10 to binary

   h. (2 pts) Convert 100 001 100 from binary to base 8

   i. (2 pts) Convert 0100 1100 0111 1010 from binary to hexadecimal

   j. (2 pts) Convert 110 011 100 from base 2 to octal

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

b. (2 pts) z->next->next

c. (2 pts) s

d. (2 pts) &a

e. (2 pts) *a

f. (2 pts) argv[0]

g. (2 pts) argc

h. (2 pts) y

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

j. (2 pts) z->data

k. (2 pts) z->next

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?

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

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

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

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
   b. (2 pts) Convert 219 from base 10 to base 2
   c. (2 pts) Convert 1100 1000 0111 0011 from binary to hexadecimal
   d. (2 pts) Convert 011 100 100 from base 2 to octal
   e. (2 pts) Convert 1111 1100 0100 1000 from binary to hexadecimal
   f. (2 pts) Convert 1000 1110 from base 2 to base 10
   g. (2 pts) Convert 1101 0001 0110 1011 from base 2 to hexadecimal
   h. (2 pts) Convert dfb from hexadecimal to base 2
   i. (2 pts) Convert 1691 from base 16 to base 2
   j. (2 pts) Convert 011 010 100 from base 2 to octal

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

b. (2 pts) x->next

c. (2 pts) *t

d. (2 pts) x->next->next

e. (2 pts) &p

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

g. (2 pts) x->data

h. (2 pts) argc

i. (2 pts) argv[0]

j. (2 pts) y

k. (2 pts) r

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);
```

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

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

Comply with these instructions for full credit:

- 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?

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

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

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

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

   b. (2 pts) Convert 1010 1100 from binary to decimal

   c. (2 pts) Convert 77 from octal to binary

   d. (2 pts) Convert ce92 from base 16 to base 2

   e. (2 pts) Convert 5266 from hexadecimal to binary

   f. (2 pts) Convert 1100 0001 0001 0100 from binary to hexadecimal

   g. (2 pts) Convert 1101 0011 from base 2 to base 10

   h. (2 pts) Convert 76ea from base 16 to base 2

   i. (2 pts) Convert 1011 1000 0100 1100 from binary to hexadecimal

   j. (2 pts) Convert 100 101 100 from base 2 to base 8

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]

b. (2 pts) argc

c. (2 pts) &d

d. (2 pts) &g

e. (2 pts) e->data

f. (2 pts) *h

g. (2 pts) argv[0]

h. (2 pts) d

i. (2 pts) e->next->next

j. (2 pts) e->next

k. (2 pts) g

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?

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

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

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

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

   b. (2 pts) Convert 0110 0110 1101 0001 from binary to base 16

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

   d. (2 pts) Convert 90 from base 10 to binary

   e. (2 pts) Convert 0011 0101 1011 0010 from binary to hexadecimal

   f. (2 pts) Convert 5af1 from base 16 to binary

   g. (2 pts) Convert 0111 1101 from binary to base 10

   h. (2 pts) Convert 111 111 101 from base 2 to base 8

   i. (2 pts) Convert 8263 from base 16 to base 2

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

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

b. (2 pts) argv[0]

c. (2 pts) c->next

d. (2 pts) &b

e. (2 pts) argc

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

g. (2 pts) c

h. (2 pts) c->next->next

i. (2 pts) c->data

j. (2 pts) y

k. (2 pts) *c

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?

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

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

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

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

   b. (2 pts) Convert 9 from base 10 to binary

   c. (2 pts) Convert 1001 0010 from binary to base 10

   d. (2 pts) Convert 5 from base 8 to binary

   e. (2 pts) Convert a62b from hexadecimal to binary

   f. (2 pts) Convert 27 from base 8 to binary

   g. (2 pts) Convert 1100 1111 0011 1101 from binary to base 16

   h. (2 pts) Convert 165 from base 10 to base 2

   i. (2 pts) Convert 74d6 from hexadecimal to binary

   j. (2 pts) Convert 17 from base 8 to binary

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

b. (2 pts) argv[0]

c. (2 pts) t->data

d. (2 pts) argc

e. (2 pts) p

f. (2 pts) t->next

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

h. (2 pts) &x

i. (2 pts) s

j. (2 pts) *s

k. (2 pts) t->next->next

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?

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

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

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

   b. (2 pts) Convert c3c6 from base 16 to base 2

   c. (2 pts) Convert 37 from base 8 to base 2

   d. (2 pts) Convert 101 000 011 from base 2 to base 8

   e. (2 pts) Convert 1000 1001 0111 0110 from binary to hexadecimal

   f. (2 pts) Convert 111 101 100 from base 2 to base 8

   g. (2 pts) Convert 0111 1001 0010 1110 from base 2 to base 16

   h. (2 pts) Convert 2cbd from hexadecimal to base 2

   i. (2 pts) Convert 0011 1110 1110 1101 from base 2 to base 16

   j. (2 pts) Convert 66 from octal to base 2

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

b. (2 pts) &q

c. (2 pts) *r

d. (2 pts) s

e. (2 pts) r->data

f. (2 pts) r->next->next

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

h. (2 pts) r->next

i. (2 pts) h

j. (2 pts) argv[0]

k. (2 pts) &f

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?

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

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

d. (2 pts) What is the value of argv[2][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 1011 0100 from binary to base 10

   b. (2 pts) Convert 1001 0101 0100 1100 from binary to hexadecimal

   c. (2 pts) Convert 182 from base 10 to base 2

   d. (2 pts) Convert 1111 1110 from base 2 to decimal

   e. (2 pts) Convert 67 from octal to binary

   f. (2 pts) Convert 0010 1000 from binary to decimal

   g. (2 pts) Convert 0111 1011 0101 1100 from binary to hexadecimal

   h. (2 pts) Convert 45 from octal to base 2

   i. (2 pts) Convert 1110 0000 1010 1000 from binary to hexadecimal

   j. (2 pts) Convert 1 from octal to binary

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

b. (2 pts) s

c. (2 pts) r->next

d. (2 pts) argv[0]

e. (2 pts) argv[1][2]

f. (2 pts) f

g. (2 pts) argc

h. (2 pts) &p

i. (2 pts) *r

j. (2 pts) r->next->next

k. (2 pts) &s

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:

- 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.

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

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

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?

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

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

d. (2 pts) What is the value of argv[1][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 f9ab from base 16 to binary

   b. (2 pts) Convert 23 from octal to base 2

   c. (2 pts) Convert 101 001 000 from base 2 to base 8

   d. (2 pts) Convert 42 from base 8 to base 2

   e. (2 pts) Convert 1100 0010 from binary to base 10

   f. (2 pts) Convert 110 000 100 from binary to octal

   g. (2 pts) Convert 0010 0101 0100 1100 from base 2 to base 16

   h. (2 pts) Convert 111 010 from base 2 to base 8

   i. (2 pts) Convert 52 from base 8 to base 2

   j. (2 pts) Convert 50 from base 8 to binary

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

b. (2 pts) argc

c. (2 pts) z->next

d. (2 pts) z->next->next

e. (2 pts) &w

f. (2 pts) z

g. (2 pts) x

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

i. (2 pts) &z

j. (2 pts) *b

k. (2 pts) argv[0]

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?

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

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

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

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
   b. (2 pts) Convert 55 from decimal to base 2
   c. (2 pts) Convert 0101 1010 0010 from binary to hexadecimal
   d. (2 pts) Convert 60 from base 8 to base 2
   e. (2 pts) Convert 1001 1100 from binary to decimal
   f. (2 pts) Convert 255 from decimal to base 2
   g. (2 pts) Convert 1011 0101 from base 2 to base 10
   h. (2 pts) Convert 185 from base 10 to base 2
   i. (2 pts) Convert 1110 0000 from binary to decimal
   j. (2 pts) Convert 001 010 101 from binary to base 8

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

b. (2 pts) s

c. (2 pts) w->data

d. (2 pts) argv[0]

e. (2 pts) r

f. (2 pts) argc

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

h. (2 pts) &q

i. (2 pts) *w

j. (2 pts) w->next->next

k. (2 pts) w->next

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?

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

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

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

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

   b. (2 pts) Convert 74 from base 8 to binary

   c. (2 pts) Convert 243 from decimal to binary

   d. (2 pts) Convert 0100 1110 1110 1001 from base 2 to hexadecimal

   e. (2 pts) Convert 800f from hexadecimal to base 2

   f. (2 pts) Convert 1001 1001 from base 2 to decimal

   g. (2 pts) Convert 5f04 from hexadecimal to base 2

   h. (2 pts) Convert 20 from octal to base 2

   i. (2 pts) Convert 1010 1010 from binary to base 10

   j. (2 pts) Convert 110 001 101 from binary to octal

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

b. (2 pts) s

c. (2 pts) r->data

d. (2 pts) *s

e. (2 pts) r->next->next

f. (2 pts) argv[0]

g. (2 pts) argc

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

i. (2 pts) &f

j. (2 pts) f

k. (2 pts) &s

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?

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

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

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

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
   b. (2 pts) Convert 1100 0011 0000 1000 from base 2 to base 16
   c. (2 pts) Convert 12 from octal to binary
   d. (2 pts) Convert 1010 1011 from binary to base 10
   e. (2 pts) Convert 1101 0110 0010 1101 from binary to base 16
   f. (2 pts) Convert 1100 1011 1001 1000 from binary to base 16
   g. (2 pts) Convert 0110 0001 from base 2 to decimal
   h. (2 pts) Convert 52 from octal to base 2
   i. (2 pts) Convert 0100 1011 from binary to decimal
   j. (2 pts) Convert f2e7 from hexadecimal to base 2

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

b. (2 pts) b->next

c. (2 pts) argc

d. (2 pts) *d

e. (2 pts) b->next->next

f. (2 pts) argv[0]

g. (2 pts) &a

h. (2 pts) d

i. (2 pts) b->data

j. (2 pts) argv[1][2]

k. (2 pts) &w

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?

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

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

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

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
   b. (2 pts) Convert 0111 1101 from binary to decimal
   c. (2 pts) Convert 101 111 from base 2 to octal
   d. (2 pts) Convert 15 from octal to binary
   e. (2 pts) Convert 56 from octal to base 2
   f. (2 pts) Convert 6575 from hexadecimal to base 2
   g. (2 pts) Convert 1011 from binary to decimal
   h. (2 pts) Convert 50 from base 10 to base 2
   i. (2 pts) Convert 21 from base 10 to base 2
   j. (2 pts) Convert 8ed1 from base 16 to base 2

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]

b. (2 pts) y

c. (2 pts) &a

d. (2 pts) &w

e. (2 pts) s

f. (2 pts) *x

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

h. (2 pts) y->data

i. (2 pts) y->next->next

j. (2 pts) y->next

k. (2 pts) argc

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?

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

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

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
   b. (2 pts) Convert 7 from octal to base 2
   c. (2 pts) Convert 1011 1101 from binary to base 10
   d. (2 pts) Convert f2a7 from base 16 to binary
   e. (2 pts) Convert 41 from decimal to binary
   f. (2 pts) Convert 31 from octal to base 2
   g. (2 pts) Convert 5cd6 from base 16 to binary
   h. (2 pts) Convert d89b from base 16 to base 2
   i. (2 pts) Convert 010 000 from base 2 to octal
   j. (2 pts) Convert 100 110 100 from base 2 to octal

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

b. (2 pts) argv[0]

c. (2 pts) h->next

d. (2 pts) &p

e. (2 pts) argc

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

g. (2 pts) h->data

h. (2 pts) *h

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

j. (2 pts) &g

k. (2 pts) h

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?

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

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

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

End of Exam

total points=100