CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

1. Please perform the following number conversions.
   a. (2 pts) Convert 0100 0110 0100 1101 from base 2 to hexadecimal

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

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

   d. (2 pts) Convert 7e91 from base 16 to base 2

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

   f. (2 pts) Convert 1010 0010 from base 2 to base 16

   g. (2 pts) Convert 0110 1100 0110 from binary to hexadecimal

   h. (2 pts) Convert 30 from base 8 to base 2

   i. (2 pts) Convert 1101 0010 from binary to base 10

   j. (2 pts) Convert 001 101 100 from base 2 to octal

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) g->next->next

c. (2 pts) g->next

d. (2 pts) c

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

f. (2 pts) &c

g. (2 pts) g->data

h. (2 pts) h

i. (2 pts) argv[0]

j. (2 pts) *f

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 date apple lime 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][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 77af from hexadecimal to base 2

   b. (2 pts) Convert 1010 1011 from base 2 to base 10

   c. (2 pts) Convert 225 from base 10 to base 2

   d. (2 pts) Convert 1101 1010 1101 0011 from base 2 to base 16

   e. (2 pts) Convert 30 from octal to base 2

   f. (2 pts) Convert 001 100 110 from base 2 to octal

   g. (2 pts) Convert 8f4 from base 16 to base 2

   h. (2 pts) Convert 62 from octal to base 2

   i. (2 pts) Convert 011 101 000 from base 2 to octal

   j. (2 pts) Convert 011 000 101 from base 2 to octal

2. Given the following declarations:

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


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


  return 0;
}
```

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

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

b. (2 pts) f->next

c. (2 pts) argc

d. (2 pts) *f

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

f. (2 pts) a

g. (2 pts) &a

h. (2 pts) &f

i. (2 pts) f->data

j. (2 pts) g

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 kiwi cherry lime
```

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

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

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 57 from octal to binary

   b. (2 pts) Convert 101 from decimal to binary

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

   d. (2 pts) Convert 0110 0110 1011 1101 from base 2 to hexadecimal

   e. (2 pts) Convert 010 001 101 from binary to octal

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

   g. (2 pts) Convert b2e5 from hexadecimal to binary

   h. (2 pts) Convert 80 from decimal to base 2

   i. (2 pts) Convert 0011 1110 from binary to base 10

   j. (2 pts) Convert 111 111 100 from binary to octal

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) &q

b. (2 pts) argv[0]

c. (2 pts) argc

d. (2 pts) &w

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

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

g. (2 pts) *t

h. (2 pts) r->next

i. (2 pts) p

j. (2 pts) s

k. (2 pts) r->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 lemon fig
```

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][4]?

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 5 from octal to binary

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

   c. (2 pts) Convert 116 from base 10 to binary

   d. (2 pts) Convert 001 000 100 from base 2 to octal

   e. (2 pts) Convert 55 from octal to base 2

   f. (2 pts) Convert ce39 from hexadecimal to base 2

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

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

   i. (2 pts) Convert 0011 0000 from base 2 to decimal

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

2. Given the following declarations:

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


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


  return 0;
}
```

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

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

b. (2 pts) argc

c. (2 pts) d

d. (2 pts) a

e. (2 pts) &c

f. (2 pts) d->next

g. (2 pts) argv[0]

h. (2 pts) d->data

i. (2 pts) *f

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

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 kiwi date
```

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[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 89 from decimal to binary

   b. (2 pts) Convert 1100 0010 1100 1000 from binary to base 16

   c. (2 pts) Convert 011 000 101 from binary to octal

   d. (2 pts) Convert 101 011 100 from base 2 to base 8

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

   f. (2 pts) Convert 104 from decimal to base 2

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

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

   i. (2 pts) Convert 111 110 101 from binary to octal

   j. (2 pts) Convert 51 from octal to base 2

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) w

b. (2 pts) argc

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

d. (2 pts) d

e. (2 pts) c->next

f. (2 pts) c->data

g. (2 pts) &a

h. (2 pts) *d

i. (2 pts) &y

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

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 date apple lemon
```

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][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 100 010 110 from base 2 to base 8

   b. (2 pts) Convert 944e from base 16 to binary

   c. (2 pts) Convert 98e4 from hexadecimal to base 2

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

   e. (2 pts) Convert 1111 0000 from binary to decimal

   f. (2 pts) Convert 46 from base 8 to binary

   g. (2 pts) Convert 1011 0000 1011 0111 from base 2 to hexadecimal

   h. (2 pts) Convert 231 from decimal to base 2

   i. (2 pts) Convert 100 111 001 from base 2 to octal

   j. (2 pts) Convert 64 from octal 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) argc

b. (2 pts) d

c. (2 pts) &d

d. (2 pts) y

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

f. (2 pts) &y

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

h. (2 pts) argv[0]

i. (2 pts) b->data

j. (2 pts) *e

k. (2 pts) b->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 kiwi lime guava
```

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][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 d00c from base 16 to base 2

   b. (2 pts) Convert 0100 1110 0111 0111 from base 2 to base 16

   c. (2 pts) Convert 86c5 from hexadecimal to binary

   d. (2 pts) Convert 150 from base 10 to base 2

   e. (2 pts) Convert d3d3 from hexadecimal to binary

   f. (2 pts) Convert 0011 0100 0101 1100 from binary to hexadecimal

   g. (2 pts) Convert 0101 1010 1010 1000 from binary to base 16

   h. (2 pts) Convert 111 from decimal to base 2

   i. (2 pts) Convert 011 001 101 from binary to base 8

   j. (2 pts) Convert 33 from base 8 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[1][2]

b. (2 pts) y

c. (2 pts) y->next

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

e. (2 pts) argv[0]

f. (2 pts) &x

g. (2 pts) *x

h. (2 pts) &r

i. (2 pts) y->data

j. (2 pts) argc

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 fig lime
```

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][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 2868 from base 16 to base 2

   b. (2 pts) Convert 241 from base 10 to binary

   c. (2 pts) Convert 44 from decimal to base 2

   d. (2 pts) Convert 24 from octal to base 2

   e. (2 pts) Convert 21 from base 8 to base 2

   f. (2 pts) Convert 001 101 011 from binary to octal

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

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

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

   j. (2 pts) Convert 0111 1001 1011 1000 from binary to hexadecimal

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) p->next

b. (2 pts) *r

c. (2 pts) &r

d. (2 pts) argv[0]

e. (2 pts) f

f. (2 pts) &d

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

h. (2 pts) r

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

j. (2 pts) p->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);
```

Comply with these instructions for full credit:

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

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

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

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

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

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

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

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

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

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

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

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

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

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

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

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

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

```
./runIt banana lime
```

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

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 011 011 010 from base 2 to octal

   b. (2 pts) Convert 52 from octal to base 2

   c. (2 pts) Convert 1a07 from hexadecimal to base 2

   d. (2 pts) Convert 1101 1101 1110 0011 from binary to hexadecimal

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

   f. (2 pts) Convert 63 from octal to base 2

   g. (2 pts) Convert 0101 0110 from binary to decimal

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

   i. (2 pts) Convert 10 from octal to binary

   j. (2 pts) Convert 0001 0101 1010 0011 from base 2 to hexadecimal

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) y->data

b. (2 pts) r

c. (2 pts) *x

d. (2 pts) &t

e. (2 pts) argc

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

g. (2 pts) &x

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

i. (2 pts) argv[0]

j. (2 pts) y

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:

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

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

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

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

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

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

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

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

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

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

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

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

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

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

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

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

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

```
./runIt banana date mango fig
```

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

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

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 1001 1110 1010 1010 from binary to hexadecimal
   b. (2 pts) Convert 0111 1100 1111 0001 from base 2 to hexadecimal
   c. (2 pts) Convert 010 100 000 from binary to octal
   d. (2 pts) Convert 0011 1010 from base 2 to decimal
   e. (2 pts) Convert 125 from base 10 to binary
   f. (2 pts) Convert 1 from base 10 to binary
   g. (2 pts) Convert 010 110 000 from base 2 to base 8
   h. (2 pts) Convert 6 from base 10 to base 2
   i. (2 pts) Convert 1100 0100 from base 2 to base 10
   j. (2 pts) Convert 20 from base 8 to base 2

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) &s

b. (2 pts) *x

c. (2 pts) &z

d. (2 pts) y->data

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

f. (2 pts) argc

g. (2 pts) argv[0]

h. (2 pts) t

i. (2 pts) y

j. (2 pts) y->next

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);
```

```
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 grape
```

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][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 1110 0011 from base 2 to base 10

   b. (2 pts) Convert 0011 0111 from base 2 to base 10

   c. (2 pts) Convert 0011 1110 0100 0110 from base 2 to hexadecimal

   d. (2 pts) Convert 61 from octal to binary

   e. (2 pts) Convert 0110 0001 0000 0000 from base 2 to base 16

   f. (2 pts) Convert 1001 1011 from base 2 to decimal

   g. (2 pts) Convert 100 from binary to octal

   h. (2 pts) Convert 142 from base 10 to base 2

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

   j. (2 pts) Convert 1101 1101 1100 1110 from binary to base 16

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) argv[0]

b. (2 pts) *w

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

d. (2 pts) s

e. (2 pts) t->data

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

g. (2 pts) &t

h. (2 pts) argc

i. (2 pts) t->next

j. (2 pts) h

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 grape mango
```

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][1]?

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

1. Please perform the following number conversions.
   a. (2 pts) Convert 1001 1111 from binary to decimal
   b. (2 pts) Convert 100 101 001 from base 2 to octal
   c. (2 pts) Convert 279c from base 16 to base 2
   d. (2 pts) Convert 4 from base 10 to base 2
   e. (2 pts) Convert 104 from decimal to binary
   f. (2 pts) Convert 011 000 011 from base 2 to base 8
   g. (2 pts) Convert 0001 0110 from binary to decimal
   h. (2 pts) Convert 17 from base 8 to binary
   i. (2 pts) Convert 62 from base 10 to base 2
   j. (2 pts) Convert 33 from octal to binary

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) &d

b. (2 pts) d

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

d. (2 pts) &x

e. (2 pts) argv[0]

f. (2 pts) b->next

g. (2 pts) argc

h. (2 pts) z

i. (2 pts) b->data

j. (2 pts) *d

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 cherry date
```

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

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

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 e4db from hexadecimal to binary

   b. (2 pts) Convert 4ef6 from hexadecimal to base 2

   c. (2 pts) Convert 0001 0101 from binary to base 10

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

   e. (2 pts) Convert 0100 1011 0101 1001 from binary to base 16

   f. (2 pts) Convert 76 from octal to base 2

   g. (2 pts) Convert 1011 1111 from base 2 to base 10

   h. (2 pts) Convert 198 from base 10 to binary

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

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

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) s

b. (2 pts) z->next

c. (2 pts) y

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

e. (2 pts) argv[0]

f. (2 pts) argc

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

h. (2 pts) z->data

i. (2 pts) &x

j. (2 pts) *a

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 banana date kiwi 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[1][1]?

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 22 from base 10 to base 2

   b. (2 pts) Convert 32 from decimal to binary

   c. (2 pts) Convert 0100 1011 1101 1011 from binary to base 16

   d. (2 pts) Convert 1110 1100 from base 2 to decimal

   e. (2 pts) Convert 50 from octal to base 2

   f. (2 pts) Convert 45 from base 10 to binary

   g. (2 pts) Convert 1100 0010 from binary to decimal

   h. (2 pts) Convert 47 from decimal to binary

   i. (2 pts) Convert 170 from base 10 to base 2

   j. (2 pts) Convert 001 101 000 from base 2 to octal

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) &w

b. (2 pts) argc

c. (2 pts) argv[0]

d. (2 pts) a->data

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

f. (2 pts) &a

g. (2 pts) w

h. (2 pts) a->next

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

j. (2 pts) y

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 guava grape fig kiwi
```

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][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 0101 1011 0001 1101 from base 2 to hexadecimal
   b. (2 pts) Convert 66 from base 8 to base 2
   c. (2 pts) Convert 0011 1001 1011 1100 from base 2 to base 16
   d. (2 pts) Convert 120 from base 10 to binary
   e. (2 pts) Convert 1000 0100 from binary to base 10
   f. (2 pts) Convert 1100 0111 from base 2 to decimal
   g. (2 pts) Convert 0110 1100 from base 2 to decimal
   h. (2 pts) Convert b723 from base 16 to binary
   i. (2 pts) Convert 748e from hexadecimal to binary
   j. (2 pts) Convert 110 100 000 from base 2 to octal

2. Given the following declarations:

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


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


  return 0;
}
```

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

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

b. (2 pts) h

c. (2 pts) a

d. (2 pts) e->next

e. (2 pts) argc

f. (2 pts) *e

g. (2 pts) &b

h. (2 pts) e->data

i. (2 pts) argv[0]

j. (2 pts) &e

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 banana cherry lemon
```

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

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

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

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 1011 0011 0111 1001 from base 2 to hexadecimal

   b. (2 pts) Convert 7d70 from base 16 to binary

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

   d. (2 pts) Convert 15 from base 8 to base 2

   e. (2 pts) Convert f53b from hexadecimal to binary

   f. (2 pts) Convert 200 from base 10 to binary

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

   h. (2 pts) Convert 010 111 011 from base 2 to octal

   i. (2 pts) Convert 0110 0111 from binary to decimal

   j. (2 pts) Convert 66 from base 8 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) y->data

b. (2 pts) argc

c. (2 pts) argv[0]

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

e. (2 pts) r

f. (2 pts) y->next

g. (2 pts) *x

h. (2 pts) &s

i. (2 pts) y

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

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);
```

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 cherry guava
```

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][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 1111 1000 from base 2 to decimal

   b. (2 pts) Convert 0011 0111 1001 1001 from base 2 to base 16

   c. (2 pts) Convert 1100 1100 1111 1110 from binary to hexadecimal

   d. (2 pts) Convert 1011 1111 1110 1110 from binary to hexadecimal

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

   f. (2 pts) Convert 98 from decimal to base 2

   g. (2 pts) Convert 67bc from base 16 to binary

   h. (2 pts) Convert 229 from base 10 to binary

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

   j. (2 pts) Convert 0111 0111 1010 1001 from binary to base 16

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) q

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

c. (2 pts) s

d. (2 pts) *t

e. (2 pts) argc

f. (2 pts) &t

g. (2 pts) &h

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

i. (2 pts) w->data

j. (2 pts) w->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);
```

```
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 grape fig 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][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 12 from octal to base 2
   b. (2 pts) Convert 91e from hexadecimal to binary
   c. (2 pts) Convert 0011 from base 2 to base 10
   d. (2 pts) Convert 0001 1100 from base 2 to base 10
   e. (2 pts) Convert 001 011 101 from binary to base 8
   f. (2 pts) Convert 1001 0101 0011 1101 from binary to hexadecimal
   g. (2 pts) Convert 0110 1001 from binary to base 10
   h. (2 pts) Convert 4e34 from hexadecimal to binary
   i. (2 pts) Convert 1101 0010 from binary to base 10
   j. (2 pts) Convert 50 from base 8 to base 2

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) w->data

b. (2 pts) &w

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

d. (2 pts) argv[0]

e. (2 pts) w->next

f. (2 pts) r

g. (2 pts) *t

h. (2 pts) s

i. (2 pts) &q

j. (2 pts) w->next->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 guava date 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][3]?

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 110 from base 10 to base 2
   b. (2 pts) Convert 110 000 110 from base 2 to base 8
   c. (2 pts) Convert 74 from base 8 to binary
   d. (2 pts) Convert 52 from base 8 to binary
   e. (2 pts) Convert 0001 0001 from binary to decimal
   f. (2 pts) Convert 2f1a from base 16 to binary
   g. (2 pts) Convert 0001 0011 from base 2 to base 10
   h. (2 pts) Convert 65 from octal to binary
   i. (2 pts) Convert 156 from base 10 to binary
   j. (2 pts) Convert 17 from octal to base 2

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) *r

b. (2 pts) p->data

c. (2 pts) s

d. (2 pts) &q

e. (2 pts) argc

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

g. (2 pts) p->next

h. (2 pts) argv[0]

i. (2 pts) &f

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

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);
```

```
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[2][0]?

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

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

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

1. Please perform the following number conversions.
   a. (2 pts) Convert 198 from decimal to base 2
   b. (2 pts) Convert 31 from base 8 to binary
   c. (2 pts) Convert 100 101 101 from binary to base 8
   d. (2 pts) Convert 63ad from hexadecimal to binary
   e. (2 pts) Convert 100 000 100 from binary to base 8
   f. (2 pts) Convert 0011 0000 from base 2 to decimal
   g. (2 pts) Convert 0110 0101 1000 1110 from base 2 to hexadecimal
   h. (2 pts) Convert 124 from decimal to binary
   i. (2 pts) Convert 143 from decimal to binary
   j. (2 pts) Convert 0100 1011 0110 1000 from binary to hexadecimal

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) &e

b. (2 pts) &y

c. (2 pts) argv[0]

d. (2 pts) y

e. (2 pts) b->data

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

g. (2 pts) b->next

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

i. (2 pts) *d

j. (2 pts) c

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 apple
```

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 2 from base 8 to binary
   b. (2 pts) Convert 001 000 000 from base 2 to octal
   c. (2 pts) Convert 1000 0100 1000 0000 from base 2 to base 16
   d. (2 pts) Convert ef97 from hexadecimal to base 2
   e. (2 pts) Convert 0110 0101 from binary to decimal
   f. (2 pts) Convert 110 010 100 from base 2 to octal
   g. (2 pts) Convert 1111 0111 1111 from binary to base 16
   h. (2 pts) Convert 4 from decimal to binary
   i. (2 pts) Convert 0101 1001 from base 2 to base 10
   j. (2 pts) Convert 1110 0111 0101 0011 from base 2 to hexadecimal

2. Given the following declarations:

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


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


  return 0;
}
```

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

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

b. (2 pts) *r

c. (2 pts) &d

d. (2 pts) argc

e. (2 pts) argv[0]

f. (2 pts) g

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

h. (2 pts) h

i. (2 pts) p->data

j. (2 pts) &q

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 date apple
```

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

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

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

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 3d38 from hexadecimal to binary
   b. (2 pts) Convert 1111 0001 from binary to decimal
   c. (2 pts) Convert 186 from base 10 to binary
   d. (2 pts) Convert 0100 1011 1101 1000 from binary to hexadecimal
   e. (2 pts) Convert 187 from decimal to binary
   f. (2 pts) Convert fcb0 from hexadecimal to base 2
   g. (2 pts) Convert 0001 0001 1010 1101 from base 2 to hexadecimal
   h. (2 pts) Convert 6cf6 from base 16 to binary
   i. (2 pts) Convert fb2f from hexadecimal to binary
   j. (2 pts) Convert 0001 0011 1001 0100 from binary to base 16

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) argc

b. (2 pts) &h

c. (2 pts) s

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

e. (2 pts) &s

f. (2 pts) argv[0]

g. (2 pts) r->next

h. (2 pts) *s

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

j. (2 pts) e

k. (2 pts) r->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);
```

```
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 apple fig
```

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][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 100 000 100 from binary to base 8

   b. (2 pts) Convert 171 from decimal to base 2

   c. (2 pts) Convert 168 from decimal to binary

   d. (2 pts) Convert 1101 0111 1100 0011 from binary to base 16

   e. (2 pts) Convert 1001 1111 0001 1101 from binary to hexadecimal

   f. (2 pts) Convert 150 from base 10 to binary

   g. (2 pts) Convert 1011 1011 1001 1110 from binary to hexadecimal

   h. (2 pts) Convert 75 from octal to binary

   i. (2 pts) Convert 1100 0101 from base 2 to base 10

   j. (2 pts) Convert 1010 1111 0111 1110 from base 2 to hexadecimal

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) *e

b. (2 pts) &c

c. (2 pts) f->next

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

e. (2 pts) &f

f. (2 pts) argc

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

h. (2 pts) c

i. (2 pts) argv[0]

j. (2 pts) f->data

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 lime guava kiwi
```

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 110 110 100 from binary to base 8

   b. (2 pts) Convert 010 011 101 from binary to base 8

   c. (2 pts) Convert 4e21 from base 16 to base 2

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

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

   f. (2 pts) Convert 1001 0111 1101 1110 from base 2 to hexadecimal

   g. (2 pts) Convert 011 010 from base 2 to octal

   h. (2 pts) Convert 155 from decimal to base 2

   i. (2 pts) Convert 1011 0111 from binary to decimal

   j. (2 pts) Convert bb12 from base 16 to base 2

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) *x

b. (2 pts) z->data

c. (2 pts) t

d. (2 pts) y

e. (2 pts) argc

f. (2 pts) &t

g. (2 pts) &x

h. (2 pts) z->next

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

j. (2 pts) argv[0]

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);
```

```
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 lime fig banana
```

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

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

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

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 31 from decimal to base 2
   b. (2 pts) Convert 8df from base 16 to binary
   c. (2 pts) Convert 0011 1100 from binary to decimal
   d. (2 pts) Convert 111 110 from binary to octal
   e. (2 pts) Convert 242 from decimal to binary
   f. (2 pts) Convert 31bb from base 16 to binary
   g. (2 pts) Convert 1011 0111 from binary to decimal
   h. (2 pts) Convert 22c8 from hexadecimal to base 2
   i. (2 pts) Convert 100 000 011 from base 2 to base 8
   j. (2 pts) Convert 56fc from base 16 to base 2

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) h

b. (2 pts) argv[0]

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

d. (2 pts) s

e. (2 pts) &p

f. (2 pts) &w

g. (2 pts) w->data

h. (2 pts) *t

i. (2 pts) w->next

j. (2 pts) argc

k. (2 pts) w->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 mango grape guava kiwi
```

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][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 0101 0000 from base 2 to decimal

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

   c. (2 pts) Convert 34 from octal to base 2

   d. (2 pts) Convert 0111 1011 from binary to decimal

   e. (2 pts) Convert 48ff from hexadecimal to binary

   f. (2 pts) Convert 31 from octal to base 2

   g. (2 pts) Convert 1011 1001 0111 0000 from base 2 to base 16

   h. (2 pts) Convert 42 from base 8 to binary

   i. (2 pts) Convert 35 from decimal to binary

   j. (2 pts) Convert 1000 0011 0011 1101 from base 2 to hexadecimal

2. Given the following declarations:

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


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


  return 0;
}
```

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

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

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

c. (2 pts) e->data

d. (2 pts) argc

e. (2 pts) *f

f. (2 pts) e->next

g. (2 pts) &e

h. (2 pts) argv[0]

i. (2 pts) h

j. (2 pts) &d

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);
```

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 cherry lime kiwi
```

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][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 9594 from base 16 to binary

   b. (2 pts) Convert 45 from octal to binary

   c. (2 pts) Convert 6040 from hexadecimal to base 2

   d. (2 pts) Convert 7 from decimal to base 2

   e. (2 pts) Convert 13 from octal to base 2

   f. (2 pts) Convert 1111 1110 0000 0001 from binary to base 16

   g. (2 pts) Convert 011 000 110 from binary to base 8

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

   i. (2 pts) Convert 1110 1101 from base 2 to base 10

   j. (2 pts) Convert 1f28 from hexadecimal to binary

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) c->next

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

c. (2 pts) &z

d. (2 pts) argc

e. (2 pts) c->data

f. (2 pts) d

g. (2 pts) argv[0]

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

i. (2 pts) x

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 date 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[2][4]?

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 1000 1001 1101 1010 from binary to base 16

   b. (2 pts) Convert 124 from decimal to binary

   c. (2 pts) Convert 110 000 011 from base 2 to base 8

   d. (2 pts) Convert 010 000 000 from base 2 to octal

   e. (2 pts) Convert 1 from base 8 to binary

   f. (2 pts) Convert 0011 1010 from binary to base 10

   g. (2 pts) Convert f327 from hexadecimal to base 2

   h. (2 pts) Convert 175 from decimal to binary

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

   j. (2 pts) Convert 52 from base 8 to binary

2. Given the following declarations:

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


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


  return 0;
}
```

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

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

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

c. (2 pts) *y

d. (2 pts) argc

e. (2 pts) &t

f. (2 pts) &z

g. (2 pts) x->data

h. (2 pts) argv[0]

i. (2 pts) y

j. (2 pts) r

k. (2 pts) x->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 fig lemon
```

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[0][2]?

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 1100 1110 from binary to decimal

   b. (2 pts) Convert 369b from base 16 to base 2

   c. (2 pts) Convert 53 from base 8 to base 2

   d. (2 pts) Convert 110 011 000 from base 2 to base 8

   e. (2 pts) Convert 1110 1001 from base 2 to decimal

   f. (2 pts) Convert 110 101 001 from base 2 to base 8

   g. (2 pts) Convert 9d18 from base 16 to binary

   h. (2 pts) Convert 15 from octal to binary

   i. (2 pts) Convert 1110 1101 0001 0001 from binary to base 16

   j. (2 pts) Convert 0100 0011 1111 0011 from binary to hexadecimal

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) argv[0]

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

c. (2 pts) f->data

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

e. (2 pts) &a

f. (2 pts) *g

g. (2 pts) argc

h. (2 pts) &d

i. (2 pts) f->next

j. (2 pts) d

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);
```

Comply with these instructions for full credit:

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

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

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

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

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

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

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

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

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

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

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

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

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

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

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

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

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

```
./runIt date lime kiwi mango
```

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

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][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 0 from base 8 to binary
   b. (2 pts) Convert 8 from base 10 to binary
   c. (2 pts) Convert 229 from base 10 to binary
   d. (2 pts) Convert 0010 1000 from base 2 to base 10
   e. (2 pts) Convert 63 from base 10 to base 2
   f. (2 pts) Convert 7 from decimal to binary
   g. (2 pts) Convert 9f46 from hexadecimal to base 2
   h. (2 pts) Convert 50 from base 8 to binary
   i. (2 pts) Convert 43 from octal to binary
   j. (2 pts) Convert 0111 0000 from base 2 to base 10

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) argv[0]

b. (2 pts) c

c. (2 pts) &f

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

e. (2 pts) f->data

f. (2 pts) &z

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

h. (2 pts) argc

i. (2 pts) *f

j. (2 pts) f->next

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 mango kiwi date
```

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][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 68 from decimal to base 2
   b. (2 pts) Convert 60 from octal to base 2
   c. (2 pts) Convert 211 from decimal to binary
   d. (2 pts) Convert 180 from base 10 to binary
   e. (2 pts) Convert 22e4 from base 16 to base 2
   f. (2 pts) Convert 161 from decimal to binary
   g. (2 pts) Convert 4936 from base 16 to base 2
   h. (2 pts) Convert 40 from base 10 to binary
   i. (2 pts) Convert 0101 1000 1110 0011 from base 2 to base 16
   j. (2 pts) Convert 3 from octal 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) argv[1][2]

b. (2 pts) y

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

d. (2 pts) &a

e. (2 pts) argv[0]

f. (2 pts) b->data

g. (2 pts) argc

h. (2 pts) d

i. (2 pts) &z

j. (2 pts) b->next

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);
```

Comply with these instructions for full credit:

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

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

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

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

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

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

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

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

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

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

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

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

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

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

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

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

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

```
./runIt apple cherry lime
```

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

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

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 157 from base 10 to base 2

   b. (2 pts) Convert 6516 from hexadecimal to binary

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

   d. (2 pts) Convert 33 from octal to binary

   e. (2 pts) Convert 147 from base 10 to binary

   f. (2 pts) Convert a262 from hexadecimal to binary

   g. (2 pts) Convert 100 110 101 from binary to base 8

   h. (2 pts) Convert 1100 1110 1010 0110 from base 2 to hexadecimal

   i. (2 pts) Convert 0100 1011 0101 0111 from base 2 to hexadecimal

   j. (2 pts) Convert 0001 0111 1011 0010 from binary to hexadecimal

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) s

b. (2 pts) s->next

c. (2 pts) argc

d. (2 pts) argv[0]

e. (2 pts) &w

f. (2 pts) &q

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

h. (2 pts) h

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

j. (2 pts) *w

k. (2 pts) s->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 guava grape
```

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][3]?

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

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

1. Please perform the following number conversions.
   a. (2 pts) Convert 1110 0010 0011 0111 from binary to base 16
   b. (2 pts) Convert 0001 1111 0011 1110 from binary to base 16
   c. (2 pts) Convert 0110 0111 from binary to decimal
   d. (2 pts) Convert 1111 1011 1101 1000 from binary to base 16
   e. (2 pts) Convert 76a8 from base 16 to base 2
   f. (2 pts) Convert 60 from base 10 to base 2
   g. (2 pts) Convert 010 001 001 from base 2 to base 8
   h. (2 pts) Convert 25 from octal to binary
   i. (2 pts) Convert 5 from octal to base 2
   j. (2 pts) Convert 1011 0011 1001 1101 from base 2 to base 16

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) h

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

c. (2 pts) argv[0]

d. (2 pts) g

e. (2 pts) &p

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

g. (2 pts) q->data

h. (2 pts) q->next

i. (2 pts) argc

j. (2 pts) *h

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 cherry mango guava banana
```

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][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 1399 from base 16 to binary

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

   c. (2 pts) Convert 47 from octal to base 2

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

   e. (2 pts) Convert 1100 1100 1100 0110 from base 2 to base 16

   f. (2 pts) Convert 33 from base 8 to binary

   g. (2 pts) Convert 4708 from hexadecimal to binary

   h. (2 pts) Convert 1011 1111 from binary to decimal

   i. (2 pts) Convert 1011 0111 0010 1000 from base 2 to hexadecimal

   j. (2 pts) Convert 1101 1111 1101 1101 from binary to hexadecimal

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) b->data

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

c. (2 pts) &t

d. (2 pts) *z

e. (2 pts) argc

f. (2 pts) argv[0]

g. (2 pts) w

h. (2 pts) b->next

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

j. (2 pts) c

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 mango lemon date fig
```

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][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 26 from base 8 to binary

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

   c. (2 pts) Convert 8b4f from base 16 to base 2

   d. (2 pts) Convert 71 from octal to base 2

   e. (2 pts) Convert 101 100 000 from base 2 to octal

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

   g. (2 pts) Convert 111 100 001 from base 2 to octal

   h. (2 pts) Convert 70 from base 10 to binary

   i. (2 pts) Convert 40 from octal to base 2

   j. (2 pts) Convert 0111 1011 1100 1000 from base 2 to hexadecimal

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) argc

b. (2 pts) &x

c. (2 pts) w->data

d. (2 pts) w->next

e. (2 pts) t

f. (2 pts) y

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

h. (2 pts) *x

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

j. (2 pts) &q

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 lime apple
```

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][5]?

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 101 100 001 from base 2 to octal

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

   c. (2 pts) Convert 0011 0000 1011 0001 from binary to hexadecimal

   d. (2 pts) Convert 9f96 from hexadecimal to base 2

   e. (2 pts) Convert 32 from base 10 to base 2

   f. (2 pts) Convert 9 from decimal to binary

   g. (2 pts) Convert 010 000 101 from binary to octal

   h. (2 pts) Convert 1110 1101 0110 1000 from base 2 to hexadecimal

   i. (2 pts) Convert 73b3 from base 16 to binary

   j. (2 pts) Convert 875c from base 16 to base 2

2. Given the following declarations:

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


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


  return 0;
}
```

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

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

b. (2 pts) &f

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

d. (2 pts) argc

e. (2 pts) h->next

f. (2 pts) *h

g. (2 pts) h->data

h. (2 pts) &q

i. (2 pts) h

j. (2 pts) argv[0]

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);
```

```
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 kiwi grape
```

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[1][0]?

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 245 from base 10 to base 2
   b. (2 pts) Convert 7 from decimal to binary
   c. (2 pts) Convert 30 from decimal to binary
   d. (2 pts) Convert 2b80 from hexadecimal to binary
   e. (2 pts) Convert 0011 1101 0101 from base 2 to base 16
   f. (2 pts) Convert 1010 0011 from binary to base 10
   g. (2 pts) Convert 111 011 001 from base 2 to octal
   h. (2 pts) Convert 35 from base 8 to binary
   i. (2 pts) Convert 0011 1101 1100 1010 from binary to base 16
   j. (2 pts) Convert 2346 from base 16 to base 2

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) argv[0]

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

c. (2 pts) &f

d. (2 pts) d

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

f. (2 pts) argc

g. (2 pts) &b

h. (2 pts) a

i. (2 pts) f->data

j. (2 pts) f->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);
```

```
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 lemon cherry fig
```

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[0][0]?

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 0010 0111 from binary to base 10
   b. (2 pts) Convert 1101 1001 from base 2 to decimal
   c. (2 pts) Convert 010 101 001 from base 2 to octal
   d. (2 pts) Convert 1000 0111 1100 0010 from binary to hexadecimal
   e. (2 pts) Convert 26 from base 8 to binary
   f. (2 pts) Convert 214 from base 10 to base 2
   g. (2 pts) Convert 111 011 101 from binary to base 8
   h. (2 pts) Convert 1101 1110 from binary to decimal
   i. (2 pts) Convert 1101 1111 1000 0101 from base 2 to hexadecimal
   j. (2 pts) Convert 0100 1111 1000 0111 from base 2 to base 16

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) g->data

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

c. (2 pts) argv[0]

d. (2 pts) *g

e. (2 pts) h

f. (2 pts) g->next

g. (2 pts) c

h. (2 pts) &b

i. (2 pts) &h

j. (2 pts) g->next->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 kiwi cherry lemon banana
```

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][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 6bf5 from hexadecimal to binary

   b. (2 pts) Convert 1001 0011 1001 0000 from base 2 to base 16

   c. (2 pts) Convert 0100 0010 1101 0001 from base 2 to hexadecimal

   d. (2 pts) Convert 19 from decimal to binary

   e. (2 pts) Convert 0011 1101 from base 2 to decimal

   f. (2 pts) Convert 6ff9 from hexadecimal to binary

   g. (2 pts) Convert 100 110 001 from base 2 to base 8

   h. (2 pts) Convert 101 from decimal to binary

   i. (2 pts) Convert 1010 1001 1001 1100 from binary to base 16

   j. (2 pts) Convert eb71 from hexadecimal to base 2

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

b. (2 pts) r->data

c. (2 pts) f

d. (2 pts) argc

e. (2 pts) s

f. (2 pts) *t

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

h. (2 pts) argv[0]

i. (2 pts) &f

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

k. (2 pts) r->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 apple
```

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

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

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

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 1100 0100 0101 0010 from binary to hexadecimal

   b. (2 pts) Convert 001 101 100 from base 2 to base 8

   c. (2 pts) Convert 72 from octal to base 2

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

   e. (2 pts) Convert adb7 from base 16 to base 2

   f. (2 pts) Convert 0111 0001 0100 1001 from binary to hexadecimal

   g. (2 pts) Convert ea6e from hexadecimal to base 2

   h. (2 pts) Convert 3 from octal to base 2

   i. (2 pts) Convert 47 from base 8 to binary

   j. (2 pts) Convert 247 from decimal to binary

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) argc

b. (2 pts) &d

c. (2 pts) c

d. (2 pts) f->next

e. (2 pts) &z

f. (2 pts) f->data

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

h. (2 pts) argv[0]

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

j. (2 pts) d

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 fig guava apple date
```

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[1][1]?

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 010 010 from binary to octal

   b. (2 pts) Convert f085 from hexadecimal to binary

   c. (2 pts) Convert 65 from base 8 to base 2

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

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

   f. (2 pts) Convert 1011 0010 0110 from binary to hexadecimal

   g. (2 pts) Convert 945f from base 16 to base 2

   h. (2 pts) Convert 1001 0011 from base 2 to base 10

   i. (2 pts) Convert 6627 from hexadecimal to base 2

   j. (2 pts) Convert 146 from decimal to binary

2. Given the following declarations:

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


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


  return 0;
}
```

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

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

b. (2 pts) *z

c. (2 pts) argc

d. (2 pts) b->next

e. (2 pts) x

f. (2 pts) argv[0]

g. (2 pts) c

h. (2 pts) &y

i. (2 pts) b->data

j. (2 pts) &b

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);
```

```
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 grape 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][5]?

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 16 from base 8 to base 2
   b. (2 pts) Convert 194 from base 10 to base 2
   c. (2 pts) Convert 12 from base 10 to base 2
   d. (2 pts) Convert 1011 0111 from binary to base 10
   e. (2 pts) Convert 111 001 110 from base 2 to octal
   f. (2 pts) Convert 001 111 011 from base 2 to base 8
   g. (2 pts) Convert 968d from hexadecimal to binary
   h. (2 pts) Convert 252 from base 10 to binary
   i. (2 pts) Convert 7e1 from base 16 to binary
   j. (2 pts) Convert bf31 from hexadecimal to binary

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) d

b. (2 pts) &a

c. (2 pts) argc

d. (2 pts) &x

e. (2 pts) c->next

f. (2 pts) c->data

g. (2 pts) argv[0]

h. (2 pts) *a

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

j. (2 pts) z

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);
```

```
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 date banana lemon
```

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][0]?

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 0111 1111 from binary to base 10
   b. (2 pts) Convert 37 from base 8 to binary
   c. (2 pts) Convert 1111 1010 from binary to base 10
   d. (2 pts) Convert 67 from decimal to binary
   e. (2 pts) Convert 1100 1010 from base 2 to base 10
   f. (2 pts) Convert 65 from base 8 to base 2
   g. (2 pts) Convert 407e from base 16 to binary
   h. (2 pts) Convert 1000 0100 0111 1001 from binary to hexadecimal
   i. (2 pts) Convert 1101 0001 1111 1000 from binary to base 16
   j. (2 pts) Convert 5b1b from base 16 to binary

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) argv[0]

b. (2 pts) &z

c. (2 pts) x->next

d. (2 pts) *y

e. (2 pts) &s

f. (2 pts) s

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

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

i. (2 pts) argc

j. (2 pts) x->data

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 cherry mango
```

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][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 0011 1011 from binary to base 10
   b. (2 pts) Convert 1101 1001 1110 0110 from base 2 to hexadecimal
   c. (2 pts) Convert 70 from octal to base 2
   d. (2 pts) Convert 81a2 from hexadecimal to base 2
   e. (2 pts) Convert 110 100 010 from base 2 to octal
   f. (2 pts) Convert 1001 1101 0010 0000 from base 2 to hexadecimal
   g. (2 pts) Convert 541d from base 16 to base 2
   h. (2 pts) Convert 15 from base 8 to base 2
   i. (2 pts) Convert 100 000 011 from binary to base 8
   j. (2 pts) Convert 1110 1001 0110 0010 from base 2 to hexadecimal

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) d

c. (2 pts) &a

d. (2 pts) &h

e. (2 pts) e->next

f. (2 pts) argv[0]

g. (2 pts) argc

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

i. (2 pts) h

j. (2 pts) *f

k. (2 pts) e->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 lemon grape
```

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][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 80c5 from base 16 to base 2
   b. (2 pts) Convert 1001 0100 from binary to decimal
   c. (2 pts) Convert d189 from base 16 to binary
   d. (2 pts) Convert d8c from hexadecimal to binary
   e. (2 pts) Convert 180 from base 10 to binary
   f. (2 pts) Convert 0011 0110 1111 1101 from binary to hexadecimal
   g. (2 pts) Convert fe0e from hexadecimal to base 2
   h. (2 pts) Convert 188 from decimal to base 2
   i. (2 pts) Convert 0100 1011 from base 2 to decimal
   j. (2 pts) Convert 854c from hexadecimal to base 2

2. Given the following declarations:

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


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


  return 0;
}
```

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

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

b. (2 pts) argv[0]

c. (2 pts) d->next

d. (2 pts) &d

e. (2 pts) *e

f. (2 pts) d->data

g. (2 pts) z

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

i. (2 pts) &a

j. (2 pts) argc

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 guava kiwi mango grape
```

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][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 178 from decimal to binary

   b. (2 pts) Convert 011 001 011 from binary to octal

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

   d. (2 pts) Convert 0110 1001 1100 1101 from binary to base 16

   e. (2 pts) Convert ae3 from base 16 to base 2

   f. (2 pts) Convert 011 010 010 from base 2 to base 8

   g. (2 pts) Convert 0000 from base 2 to base 10

   h. (2 pts) Convert 37 from base 10 to base 2

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

   j. (2 pts) Convert 1011 0001 1000 1101 from binary to hexadecimal

2. Given the following declarations:

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


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


  return 0;
}
```

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

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

b. (2 pts) q

c. (2 pts) argv[0]

d. (2 pts) &p

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

f. (2 pts) s->next

g. (2 pts) argc

h. (2 pts) s->data

i. (2 pts) *t

j. (2 pts) &w

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 lime kiwi banana
```

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][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 1111 0111 0000 0110 from binary to hexadecimal
   b. (2 pts) Convert 1fbd from base 16 to base 2
   c. (2 pts) Convert 1111 0101 from base 2 to base 10
   d. (2 pts) Convert 245 from decimal to binary
   e. (2 pts) Convert 1110 1110 0010 1110 from base 2 to base 16
   f. (2 pts) Convert 0 from base 8 to binary
   g. (2 pts) Convert aa2d from hexadecimal to base 2
   h. (2 pts) Convert 53 from base 8 to base 2
   i. (2 pts) Convert 55 from octal to binary
   j. (2 pts) Convert 0100 1101 0111 1000 from base 2 to hexadecimal

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) argc

b. (2 pts) e

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

d. (2 pts) &e

e. (2 pts) h

f. (2 pts) h->data

g. (2 pts) h->next

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

i. (2 pts) &p

j. (2 pts) *h

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 kiwi
```

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][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 23 from octal to binary

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

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

   d. (2 pts) Convert 101 100 010 from binary to octal

   e. (2 pts) Convert 5ea7 from base 16 to binary

   f. (2 pts) Convert 0100 from base 2 to decimal

   g. (2 pts) Convert 1111 1011 1110 0000 from binary to hexadecimal

   h. (2 pts) Convert 0101 0011 from base 2 to decimal

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

   j. (2 pts) Convert 590c from hexadecimal to base 2

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) argc

b. (2 pts) a->data

c. (2 pts) &a

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

e. (2 pts) &x

f. (2 pts) d

g. (2 pts) *d

h. (2 pts) argv[0]

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

j. (2 pts) a->next

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 kiwi banana
```

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][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 1001 0100 from binary to base 10
   b. (2 pts) Convert 124 from base 10 to binary
   c. (2 pts) Convert 100 010 010 from base 2 to octal
   d. (2 pts) Convert 001 111 010 from binary to base 8
   e. (2 pts) Convert 0100 0001 1111 0010 from base 2 to base 16
   f. (2 pts) Convert 100 111 100 from binary to octal
   g. (2 pts) Convert 1010 0101 1101 0000 from base 2 to hexadecimal
   h. (2 pts) Convert 219 from base 10 to binary
   i. (2 pts) Convert 011 100 111 from base 2 to octal
   j. (2 pts) Convert f4f6 from base 16 to base 2

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) y

b. (2 pts) *y

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

d. (2 pts) argv[0]

e. (2 pts) z->next

f. (2 pts) &r

g. (2 pts) z->data

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

i. (2 pts) argc

j. (2 pts) w

k. (2 pts) &z

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 kiwi lime cherry
```

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

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

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

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 110 001 011 from base 2 to octal

   b. (2 pts) Convert 4e37 from hexadecimal to base 2

   c. (2 pts) Convert 1011 1111 0110 1001 from base 2 to hexadecimal

   d. (2 pts) Convert 1001 1001 from binary to decimal

   e. (2 pts) Convert 9810 from base 16 to base 2

   f. (2 pts) Convert d0da from hexadecimal to base 2

   g. (2 pts) Convert a7fe from base 16 to binary

   h. (2 pts) Convert 68 from decimal to base 2

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

   j. (2 pts) Convert 2137 from base 16 to base 2

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) g->next

b. (2 pts) argv[0]

c. (2 pts) &e

d. (2 pts) &g

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

f. (2 pts) c

g. (2 pts) g->data

h. (2 pts) argc

i. (2 pts) h

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

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 fig banana 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][5]?

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 10 from decimal to binary

   b. (2 pts) Convert 1000 0110 0000 from binary to hexadecimal

   c. (2 pts) Convert 173 from decimal to binary

   d. (2 pts) Convert 37 from decimal to base 2

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

   f. (2 pts) Convert 106 from base 10 to base 2

   g. (2 pts) Convert 51ef from hexadecimal to binary

   h. (2 pts) Convert 62 from base 8 to base 2

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

   j. (2 pts) Convert bd21 from base 16 to base 2

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) argc

b. (2 pts) e->data

c. (2 pts) argv[0]

d. (2 pts) &f

e. (2 pts) *f

f. (2 pts) d

g. (2 pts) b

h. (2 pts) &b

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

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

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);
```

```
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 kiwi grape 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[2][4]?

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

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

1. Please perform the following number conversions.
   a. (2 pts) Convert 62e0 from base 16 to binary
   b. (2 pts) Convert ab2c from base 16 to base 2
   c. (2 pts) Convert 52ab from base 16 to base 2
   d. (2 pts) Convert 70 from base 8 to binary
   e. (2 pts) Convert ebd4 from base 16 to base 2
   f. (2 pts) Convert 011 011 000 from base 2 to base 8
   g. (2 pts) Convert 101 000 111 from binary to base 8
   h. (2 pts) Convert 114 from base 10 to binary
   i. (2 pts) Convert 1101 0010 0100 0001 from binary to base 16
   j. (2 pts) Convert 200 from base 10 to binary

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) argc

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

c. (2 pts) s

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

e. (2 pts) x->data

f. (2 pts) argv[0]

g. (2 pts) *x

h. (2 pts) x->next

i. (2 pts) y

j. (2 pts) &x

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 kiwi apple
```

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][3]?

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

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

1. Please perform the following number conversions.
   a. (2 pts) Convert 51 from octal to binary
   b. (2 pts) Convert 011 001 010 from base 2 to octal
   c. (2 pts) Convert 0100 0000 from binary to base 10
   d. (2 pts) Convert 0110 1100 1101 1101 from base 2 to hexadecimal
   e. (2 pts) Convert 63 from base 8 to base 2
   f. (2 pts) Convert 1 from base 8 to binary
   g. (2 pts) Convert 010 011 011 from base 2 to octal
   h. (2 pts) Convert 250 from base 10 to binary
   i. (2 pts) Convert 47 from base 8 to binary
   j. (2 pts) Convert 100 from decimal to binary

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) *w

b. (2 pts) t->data

c. (2 pts) t->next

d. (2 pts) s

e. (2 pts) &w

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

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

h. (2 pts) argv[0]

i. (2 pts) g

j. (2 pts) &g

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 kiwi mango 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][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 1001 0010 0001 from base 2 to base 16
   b. (2 pts) Convert 15 from octal to base 2
   c. (2 pts) Convert 0111 0110 1110 1011 from binary to base 16
   d. (2 pts) Convert 1100 1001 from binary to decimal
   e. (2 pts) Convert 001 001 010 from base 2 to octal
   f. (2 pts) Convert 0011 1000 from binary to base 10
   g. (2 pts) Convert 010 011 111 from binary to base 8
   h. (2 pts) Convert 30 from octal to binary
   i. (2 pts) Convert 17 from octal to base 2
   j. (2 pts) Convert 144 from decimal to binary

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) &r

b. (2 pts) argc

c. (2 pts) s

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

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

f. (2 pts) &s

g. (2 pts) s->next

h. (2 pts) p

i. (2 pts) *w

j. (2 pts) s->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);
```

```
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 fig lemon
```

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][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 111 100 from base 2 to octal

   b. (2 pts) Convert 111 100 010 from binary to octal

   c. (2 pts) Convert 31 from base 8 to base 2

   d. (2 pts) Convert 25 from base 8 to base 2

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

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

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

   h. (2 pts) Convert 72 from octal to binary

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

   j. (2 pts) Convert 44 from decimal to binary

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) z->next

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

c. (2 pts) argc

d. (2 pts) z->data

e. (2 pts) y

f. (2 pts) &c

g. (2 pts) argv[0]

h. (2 pts) *z

i. (2 pts) &x

j. (2 pts) c

k. (2 pts) z->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);
```

```
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 cherry 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[2][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 011 101 100 from base 2 to octal

   b. (2 pts) Convert 1001 0011 1100 1111 from binary to base 16

   c. (2 pts) Convert 1010 0010 1101 from binary to hexadecimal

   d. (2 pts) Convert 109c from base 16 to base 2

   e. (2 pts) Convert 0111 1001 0000 0010 from base 2 to base 16

   f. (2 pts) Convert 211 from base 10 to base 2

   g. (2 pts) Convert 0100 1011 from binary to decimal

   h. (2 pts) Convert 145 from decimal to binary

   i. (2 pts) Convert 76 from octal to base 2

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

2. Given the following declarations:

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


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


  return 0;
}
```

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

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

b. (2 pts) q

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

d. (2 pts) &p

e. (2 pts) s->data

f. (2 pts) s->next

g. (2 pts) t

h. (2 pts) &s

i. (2 pts) argv[0]

j. (2 pts) argc

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 grape banana
```

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][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 bb3c from base 16 to binary
   b. (2 pts) Convert 0100 1101 from base 2 to base 10
   c. (2 pts) Convert 1111 1000 0000 1101 from base 2 to hexadecimal
   d. (2 pts) Convert 9c86 from base 16 to binary
   e. (2 pts) Convert 010 111 000 from base 2 to octal
   f. (2 pts) Convert 0110 1101 from base 2 to decimal
   g. (2 pts) Convert f555 from base 16 to base 2
   h. (2 pts) Convert 0001 1000 1101 0100 from binary to hexadecimal
   i. (2 pts) Convert 1100 0100 1011 0101 from base 2 to base 16
   j. (2 pts) Convert 1101 0100 from base 2 to base 10

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) &g

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

c. (2 pts) argv[0]

d. (2 pts) &d

e. (2 pts) *h

f. (2 pts) h

g. (2 pts) p->next

h. (2 pts) d

i. (2 pts) argc

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

k. (2 pts) p->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);
```

```
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 banana kiwi grape
```

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][4]?

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 1110 1100 from binary to base 10

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

   c. (2 pts) Convert 0010 1110 from base 2 to base 10

   d. (2 pts) Convert 1111 1000 1100 1000 from base 2 to base 16

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

   f. (2 pts) Convert 1001 1111 1100 0001 from binary to hexadecimal

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

   h. (2 pts) Convert 40 from base 8 to binary

   i. (2 pts) Convert 011 001 100 from binary to base 8

   j. (2 pts) Convert 0 from base 10 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) f

c. (2 pts) p->data

d. (2 pts) h

e. (2 pts) argv[0]

f. (2 pts) argc

g. (2 pts) &p

h. (2 pts) &g

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

j. (2 pts) *h

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 lime grape banana
```

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][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 0011 0001 0111 1110 from binary to base 16
   b. (2 pts) Convert d9a7 from hexadecimal to binary
   c. (2 pts) Convert 7 from base 8 to binary
   d. (2 pts) Convert 132 from base 10 to binary
   e. (2 pts) Convert 149 from base 10 to base 2
   f. (2 pts) Convert 399e from hexadecimal to base 2
   g. (2 pts) Convert 1010 0001 from base 2 to decimal
   h. (2 pts) Convert 1001 from binary to decimal
   i. (2 pts) Convert 14 from base 8 to binary
   j. (2 pts) Convert 156 from base 10 to base 2

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) &h

b. (2 pts) argc

c. (2 pts) *g

d. (2 pts) &d

e. (2 pts) h

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

g. (2 pts) f->data

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

i. (2 pts) f->next

j. (2 pts) argv[0]

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);
```

Comply with these instructions for full credit:

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

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

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

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

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

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

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

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

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

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

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

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

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

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

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

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

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

```
./runIt lemon mango kiwi
```

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

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

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 11 from base 8 to binary

   b. (2 pts) Convert 110 000 011 from binary to base 8

   c. (2 pts) Convert 7dba from hexadecimal to base 2

   d. (2 pts) Convert bd8b from hexadecimal to binary

   e. (2 pts) Convert 33 from base 8 to binary

   f. (2 pts) Convert 35 from octal to binary

   g. (2 pts) Convert 14 from base 8 to binary

   h. (2 pts) Convert 1010 0101 1111 0000 from binary to base 16

   i. (2 pts) Convert 65de from base 16 to base 2

   j. (2 pts) Convert 2555 from hexadecimal to binary

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) r->next

b. (2 pts) &q

c. (2 pts) argv[0]

d. (2 pts) g

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

f. (2 pts) *q

g. (2 pts) t

h. (2 pts) &h

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

j. (2 pts) r->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 lemon apple
```

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

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

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

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

End of Exam

total points=100