CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

1. Please perform the following number conversions.
   a. (2 pts) Convert 0110 1010 from base 2 to decimal
   b. (2 pts) Convert 0111 1011 from base 2 to decimal
   c. (2 pts) Convert 0110 1011 from binary to base 10
   d. (2 pts) Convert 0100 1001 from binary to base 10
   e. (2 pts) Convert 010 100 101 from base 2 to octal
   f. (2 pts) Convert 0001 0000 0110 1000 from base 2 to base 16
   g. (2 pts) Convert 110 110 110 from binary to octal
   h. (2 pts) Convert 0010 1101 1010 0001 from binary to base 16
   i. (2 pts) Convert 47 from decimal to binary
   j. (2 pts) Convert c140 from base 16 to binary

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) &d

b. (2 pts) argv[0]

c. (2 pts) p->data

d. (2 pts) f

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

f. (2 pts) *p

g. (2 pts) &p

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

i. (2 pts) argc

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);
```

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

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

Comply with these instructions for full credit:

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

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

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

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

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

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

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

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

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

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

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

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

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

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

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

```
./runIt lemon grape apple banana
```

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

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

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

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

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

1. Please perform the following number conversions.
   a. (2 pts) Convert 100 111 000 from binary to base 8

   b. (2 pts) Convert 77 from base 10 to binary

   c. (2 pts) Convert 1010 0001 1111 1010 from binary to base 16

   d. (2 pts) Convert 1010 0101 1011 0111 from binary to hexadecimal

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

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

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

   h. (2 pts) Convert 45 from base 8 to base 2

   i. (2 pts) Convert d1b0 from hexadecimal to base 2

   j. (2 pts) Convert ed81 from base 16 to base 2

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) *h

b. (2 pts) argc

c. (2 pts) c

d. (2 pts) &q

e. (2 pts) argv[0]

f. (2 pts) &c

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

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

i. (2 pts) h

j. (2 pts) h->next

k. (2 pts) h->data

3. (10 pts) In 24 hour time:

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

   Assume the following struct:

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

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

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

   Comply with these instructions for full credit:

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

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

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

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

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

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

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

Comply with these instructions for full credit:

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

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

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

```
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 guava 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[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 224 from decimal to base 2
   b. (2 pts) Convert 763 from hexadecimal to base 2
   c. (2 pts) Convert 8fda from base 16 to base 2
   d. (2 pts) Convert 49 from decimal to base 2
   e. (2 pts) Convert 1000 1100 from base 2 to base 10
   f. (2 pts) Convert 110 111 001 from binary to octal
   g. (2 pts) Convert 41 from octal to binary
   h. (2 pts) Convert 0001 1110 from binary to base 10
   i. (2 pts) Convert 155 from base 10 to binary
   j. (2 pts) Convert 896b from base 16 to binary

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) &c

b. (2 pts) d->data

c. (2 pts) *g

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

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

f. (2 pts) argv[0]

g. (2 pts) &e

h. (2 pts) argc

i. (2 pts) d->next

j. (2 pts) c

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 date guava
```

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

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

c. (2 pts) What is the value of argv[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 57 from decimal to base 2

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

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

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

   e. (2 pts) Convert 77 from base 8 to binary

   f. (2 pts) Convert 221 from decimal to binary

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

   h. (2 pts) Convert 1100 0100 1011 0010 from binary to base 16

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

   j. (2 pts) Convert 148 from base 10 to base 2

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) w->data

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

c. (2 pts) w->next

d. (2 pts) y

e. (2 pts) &t

f. (2 pts) argv[0]

g. (2 pts) s

h. (2 pts) argc

i. (2 pts) &w

j. (2 pts) *y

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

3. (10 pts) In 24 hour time:

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

   Assume the following struct:

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

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

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

   Comply with these instructions for full credit:

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

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

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

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

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

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

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

Comply with these instructions for full credit:

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

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

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

Comply with these instructions for full credit:

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

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

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

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

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

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

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

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

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

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

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

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

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

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

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

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

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

```
./runIt date lime
```

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

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

c. (2 pts) What is the value of argv[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 37 from octal to binary

   b. (2 pts) Convert 0110 0100 0101 0111 from binary to hexadecimal

   c. (2 pts) Convert 0010 0011 0001 1100 from base 2 to hexadecimal

   d. (2 pts) Convert 011 110 010 from base 2 to octal

   e. (2 pts) Convert 1110 0000 from base 2 to base 10

   f. (2 pts) Convert 0111 0111 from base 2 to decimal

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

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

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

   j. (2 pts) Convert 48 from decimal to base 2

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) argv[0]

b. (2 pts) *c

c. (2 pts) &e

d. (2 pts) e->data

e. (2 pts) y

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

g. (2 pts) e->next

h. (2 pts) d

i. (2 pts) &a

j. (2 pts) argc

k. (2 pts) e->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 grape fig mango lime
```

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][3]?

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

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

1. Please perform the following number conversions.

   a. (2 pts) Convert af82 from base 16 to base 2

   b. (2 pts) Convert 0011 0101 1101 1101 from base 2 to base 16

   c. (2 pts) Convert 26 from base 8 to base 2

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

   e. (2 pts) Convert 54 from base 10 to base 2

   f. (2 pts) Convert 1010 1010 0100 0101 from base 2 to base 16

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

   h. (2 pts) Convert 1011 0101 from base 2 to base 10

   i. (2 pts) Convert 250 from base 10 to base 2

   j. (2 pts) Convert 93 from decimal to binary

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) c->next

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

c. (2 pts) c

d. (2 pts) argv[0]

e. (2 pts) c->data

f. (2 pts) argc

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

h. (2 pts) &c

i. (2 pts) *f

j. (2 pts) &z

k. (2 pts) a

3. (10 pts) In 24 hour time:

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

   Assume the following struct:

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

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

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

   Comply with these instructions for full credit:

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

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

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

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

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

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

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

Comply with these instructions for full credit:

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

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

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

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

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

Comply with these instructions for full credit:

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

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

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

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

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

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

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

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

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

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

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

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

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

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

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

```
./runIt cherry guava lemon
```

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

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

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

d. (2 pts) What is the value of argv[1][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 111 101 000 from binary to base 8
   b. (2 pts) Convert 111 100 000 from base 2 to octal
   c. (2 pts) Convert 21 from octal to binary
   d. (2 pts) Convert 97 from base 10 to base 2
   e. (2 pts) Convert 0001 1001 0111 1101 from base 2 to hexadecimal
   f. (2 pts) Convert 4422 from hexadecimal to base 2
   g. (2 pts) Convert 001 011 110 from binary to base 8
   h. (2 pts) Convert 0011 1101 from base 2 to base 10
   i. (2 pts) Convert 1100 0100 from binary to decimal
   j. (2 pts) Convert 249 from decimal to binary

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) argv[0]

b. (2 pts) z->next

c. (2 pts) *a

d. (2 pts) c

e. (2 pts) z->data

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

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

h. (2 pts) t

i. (2 pts) argc

j. (2 pts) &t

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 fig guava
```

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][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 010 011 001 from base 2 to base 8
   b. (2 pts) Convert 92d2 from hexadecimal to base 2
   c. (2 pts) Convert 111 011 001 from base 2 to base 8
   d. (2 pts) Convert 1cdd from base 16 to base 2
   e. (2 pts) Convert 100 010 011 from base 2 to base 8
   f. (2 pts) Convert 010 001 010 from binary to base 8
   g. (2 pts) Convert 80c0 from base 16 to binary
   h. (2 pts) Convert 1110 0011 0111 0011 from base 2 to hexadecimal
   i. (2 pts) Convert 1011 0110 from base 2 to decimal
   j. (2 pts) Convert 0100 from binary to base 10

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) s

b. (2 pts) p

c. (2 pts) argv[0]

d. (2 pts) &p

e. (2 pts) argc

f. (2 pts) &r

g. (2 pts) *q

h. (2 pts) r->data

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

j. (2 pts) r->next

k. (2 pts) r->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 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[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 919d from hexadecimal to base 2

   b. (2 pts) Convert 010 011 001 from binary to octal

   c. (2 pts) Convert 66 from base 8 to base 2

   d. (2 pts) Convert 1010 1000 1100 0111 from base 2 to hexadecimal

   e. (2 pts) Convert 109 from base 10 to base 2

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

   g. (2 pts) Convert 2ab0 from hexadecimal to base 2

   h. (2 pts) Convert 0110 1011 from binary to decimal

   i. (2 pts) Convert 128 from decimal to base 2

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

2. Given the following declarations:

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


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


  return 0;
}
```

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

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

b. (2 pts) p->data

c. (2 pts) p->next

d. (2 pts) argv[0]

e. (2 pts) h

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

g. (2 pts) &e

h. (2 pts) *p

i. (2 pts) argc

j. (2 pts) e

k. (2 pts) &q

3. (10 pts) In 24 hour time:

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

   Assume the following struct:

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

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

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

   Comply with these instructions for full credit:

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

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

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

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

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

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

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

Comply with these instructions for full credit:

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

5. (10 pts) Given the struct definitions shown below (which are exactly as 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 apple grape guava
```

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][1]?

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

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

1. Please perform the following number conversions.
   a. (2 pts) Convert 1100 0010 from base 2 to decimal
   b. (2 pts) Convert 7 from base 8 to base 2
   c. (2 pts) Convert 16 from base 10 to binary
   d. (2 pts) Convert 1 from base 8 to base 2
   e. (2 pts) Convert c35f from base 16 to base 2
   f. (2 pts) Convert 100 011 from base 2 to base 8
   g. (2 pts) Convert 44 from base 10 to binary
   h. (2 pts) Convert 1101 0100 from binary to decimal
   i. (2 pts) Convert 34 from base 10 to base 2
   j. (2 pts) Convert 204 from base 10 to base 2

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) z->data

b. (2 pts) *a

c. (2 pts) &a

d. (2 pts) z

e. (2 pts) z->next

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

g. (2 pts) argc

h. (2 pts) argv[0]

i. (2 pts) y

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

k. (2 pts) &w

3. (10 pts) In 24 hour time:

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

   Assume the following struct:

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

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

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

   Comply with these instructions for full credit:

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

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

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

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

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

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

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

Comply with these instructions for full credit:

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

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

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

Comply with these instructions for full credit:

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

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

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

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

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

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

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

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

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

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

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

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

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

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

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

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

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

```
./runIt date grape 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[2][0]?

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

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

1. Please perform the following number conversions.
   a. (2 pts) Convert 0111 1101 1111 from base 2 to base 16
   b. (2 pts) Convert 216 from base 10 to binary
   c. (2 pts) Convert 111 111 101 from binary to octal
   d. (2 pts) Convert 44 from base 8 to binary
   e. (2 pts) Convert 1010 0110 1010 1010 from base 2 to hexadecimal
   f. (2 pts) Convert 1010 1011 from binary to base 10
   g. (2 pts) Convert 214 from decimal to binary
   h. (2 pts) Convert 0101 1011 1100 0011 from base 2 to hexadecimal
   i. (2 pts) Convert 236 from decimal to base 2
   j. (2 pts) Convert 104 from base 10 to base 2

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) z

b. (2 pts) r

c. (2 pts) *z

d. (2 pts) argc

e. (2 pts) &z

f. (2 pts) w->data

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

h. (2 pts) &q

i. (2 pts) w->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 lemon 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[1][3]?

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

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

1. Please perform the following number conversions.
   a. (2 pts) Convert 0110 0000 0011 1011 from base 2 to base 16
   b. (2 pts) Convert 011 110 110 from base 2 to octal
   c. (2 pts) Convert 1010 0100 from base 2 to base 10
   d. (2 pts) Convert 4c85 from base 16 to binary
   e. (2 pts) Convert 0001 0111 from base 2 to base 10
   f. (2 pts) Convert ace6 from hexadecimal to base 2
   g. (2 pts) Convert 12 from base 8 to base 2
   h. (2 pts) Convert 0010 0011 0101 from base 2 to hexadecimal
   i. (2 pts) Convert 1101 1110 from base 2 to decimal
   j. (2 pts) Convert 0111 0100 from base 2 to decimal

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) h->next

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

c. (2 pts) c

d. (2 pts) h->data

e. (2 pts) argv[0]

f. (2 pts) argc

g. (2 pts) *p

h. (2 pts) &c

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

j. (2 pts) &p

k. (2 pts) h

3. (10 pts) In 24 hour time:

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

   Assume the following struct:

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

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

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

   Comply with these instructions for full credit:

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

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

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

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

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

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

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

Comply with these instructions for full credit:

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

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

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

Comply with these instructions for full credit:

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

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

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

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

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

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

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

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

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

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

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

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

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

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

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

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

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

```
./runIt banana lemon
```

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

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

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

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

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

1. Please perform the following number conversions.
   a. (2 pts) Convert 101 001 010 from base 2 to base 8

   b. (2 pts) Convert 0011 0101 from binary to base 10

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

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

   e. (2 pts) Convert 250 from base 10 to binary

   f. (2 pts) Convert 70 from decimal to binary

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

   h. (2 pts) Convert 1000 1001 from base 2 to decimal

   i. (2 pts) Convert 1010 1001 from binary to base 10

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

2. Given the following declarations:

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


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


  return 0;
}
```

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

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

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

c. (2 pts) e->next

d. (2 pts) &c

e. (2 pts) *c

f. (2 pts) b

g. (2 pts) c

h. (2 pts) argv[0]

i. (2 pts) e->data

j. (2 pts) &b

k. (2 pts) argc

3. (10 pts) In 24 hour time:

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

   Assume the following struct:

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

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

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

   Comply with these instructions for full credit:

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

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

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

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

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

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

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

Comply with these instructions for full credit:

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

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

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

Comply with these instructions for full credit:

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

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

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

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

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

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

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

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

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

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

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

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

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

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

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

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

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

```
./runIt grape guava apple cherry
```

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][4]?

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 65 from octal to binary
   b. (2 pts) Convert 723 from hexadecimal to base 2
   c. (2 pts) Convert c87e from hexadecimal to base 2
   d. (2 pts) Convert 0011 0100 1011 0001 from base 2 to base 16
   e. (2 pts) Convert 508c from base 16 to binary
   f. (2 pts) Convert 36 from base 8 to binary
   g. (2 pts) Convert d4a1 from base 16 to base 2
   h. (2 pts) Convert 1111 0010 from binary to decimal
   i. (2 pts) Convert 0100 1010 from base 2 to decimal
   j. (2 pts) Convert 0011 1100 from base 2 to decimal

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) argc

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

c. (2 pts) &w

d. (2 pts) &r

e. (2 pts) t

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

g. (2 pts) w->next

h. (2 pts) p

i. (2 pts) *y

j. (2 pts) w->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 guava lime 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][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 0001 1011 from binary to decimal

   b. (2 pts) Convert c14c from base 16 to binary

   c. (2 pts) Convert b65f from base 16 to base 2

   d. (2 pts) Convert 192 from base 10 to base 2

   e. (2 pts) Convert 14 from base 8 to base 2

   f. (2 pts) Convert 0001 0011 0000 1001 from base 2 to base 16

   g. (2 pts) Convert 7e92 from hexadecimal to base 2

   h. (2 pts) Convert 0111 1010 1000 0100 from binary to hexadecimal

   i. (2 pts) Convert 20 from decimal to base 2

   j. (2 pts) Convert 1101 1000 from binary to decimal

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->next

c. (2 pts) h

d. (2 pts) &t

e. (2 pts) argv[0]

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

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

h. (2 pts) argc

i. (2 pts) &g

j. (2 pts) t

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 lime apple banana
```

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][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 1101 0111 from base 2 to base 10

   b. (2 pts) Convert 0001 1110 1111 1111 from base 2 to base 16

   c. (2 pts) Convert 159 from base 10 to binary

   d. (2 pts) Convert 77 from base 8 to binary

   e. (2 pts) Convert 0011 1010 1110 0101 from base 2 to hexadecimal

   f. (2 pts) Convert d8bd from base 16 to base 2

   g. (2 pts) Convert 9231 from hexadecimal to binary

   h. (2 pts) Convert 12 from octal to base 2

   i. (2 pts) Convert 1100 0100 0110 1100 from binary to hexadecimal

   j. (2 pts) Convert 102 from base 10 to base 2

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) &d

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

c. (2 pts) h

d. (2 pts) *p

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

f. (2 pts) &p

g. (2 pts) q->data

h. (2 pts) q->next

i. (2 pts) d

j. (2 pts) argc

k. (2 pts) argv[0]

3. (10 pts) In 24 hour time:

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

   Assume the following struct:

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

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

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

   Comply with these instructions for full credit:

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

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

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

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

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

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

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

Comply with these instructions for full credit:

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

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

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

Comply with these instructions for full credit:

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

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

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

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

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

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

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

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

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

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

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

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

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

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

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

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

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

```
./runIt grape banana date mango
```

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

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][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 0001 1100 1010 1110 from binary to hexadecimal
   b. (2 pts) Convert 66 from octal to binary
   c. (2 pts) Convert 100 011 011 from binary to octal
   d. (2 pts) Convert 1000 1010 1101 0010 from base 2 to base 16
   e. (2 pts) Convert 0001 1110 from base 2 to decimal
   f. (2 pts) Convert 114 from decimal to binary
   g. (2 pts) Convert 17 from base 8 to binary
   h. (2 pts) Convert 1011 0010 from binary to base 10
   i. (2 pts) Convert 8e83 from base 16 to base 2
   j. (2 pts) Convert 2 from base 10 to base 2

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) z

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

c. (2 pts) &t

d. (2 pts) argv[0]

e. (2 pts) &z

f. (2 pts) *y

g. (2 pts) t

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

i. (2 pts) y->next

j. (2 pts) argc

k. (2 pts) y->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 date grape apple 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 78 from base 10 to base 2

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

   c. (2 pts) Convert c3f4 from hexadecimal to base 2

   d. (2 pts) Convert 71 from base 8 to binary

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

   f. (2 pts) Convert a503 from base 16 to base 2

   g. (2 pts) Convert 3e50 from hexadecimal to base 2

   h. (2 pts) Convert 27 from base 10 to base 2

   i. (2 pts) Convert 0011 0000 0011 1110 from base 2 to hexadecimal

   j. (2 pts) Convert 46 from base 10 to binary

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) argv[0]

b. (2 pts) *y

c. (2 pts) w->next

d. (2 pts) &w

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

f. (2 pts) w->data

g. (2 pts) argc

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

i. (2 pts) y

j. (2 pts) &s

k. (2 pts) q

3. (10 pts) In 24 hour time:

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

   Assume the following struct:

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

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

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

   Comply with these instructions for full credit:

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

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

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

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

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

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

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

Comply with these instructions for full credit:

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

5. (10 pts) Given the struct definitions shown below (which are exactly as 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 lemon
```

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

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

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

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

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

1. Please perform the following number conversions.

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

   b. (2 pts) Convert 0110 0100 from base 2 to decimal

   c. (2 pts) Convert 1011 0001 from binary to decimal

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

   e. (2 pts) Convert 87 from base 10 to base 2

   f. (2 pts) Convert 0011 1110 1110 0000 from base 2 to base 16

   g. (2 pts) Convert e840 from base 16 to binary

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

   i. (2 pts) Convert fa55 from hexadecimal to base 2

   j. (2 pts) Convert 202 from decimal to binary

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) s->next

b. (2 pts) s->data

c. (2 pts) argv[0]

d. (2 pts) argc

e. (2 pts) p

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

g. (2 pts) s

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

i. (2 pts) &h

j. (2 pts) *w

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);
```

```
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 guava
```

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

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

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

d. (2 pts) What is the value of argv[0][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 72 from base 8 to base 2

   b. (2 pts) Convert 1 from base 8 to base 2

   c. (2 pts) Convert 87 from decimal to base 2

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

   e. (2 pts) Convert 110 010 000 from base 2 to octal

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

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

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

   i. (2 pts) Convert ecc8 from hexadecimal to base 2

   j. (2 pts) Convert 1101 0110 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) d

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

c. (2 pts) argv[0]

d. (2 pts) argc

e. (2 pts) f->data

f. (2 pts) b

g. (2 pts) &a

h. (2 pts) &e

i. (2 pts) *g

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

k. (2 pts) f->next

3. (10 pts) In 24 hour time:

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

   Assume the following struct:

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

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

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

   Comply with these instructions for full credit:

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

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

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

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

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

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

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

Comply with these instructions for full credit:

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

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

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

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

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

Comply with these instructions for full credit:

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

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

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

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

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

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

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

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

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

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

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

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

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

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

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

```
./runIt cherry kiwi mango 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][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 48 from base 10 to base 2

   b. (2 pts) Convert 193 from base 10 to base 2

   c. (2 pts) Convert 4516 from hexadecimal to base 2

   d. (2 pts) Convert 1011 1010 from base 2 to decimal

   e. (2 pts) Convert 171 from base 10 to base 2

   f. (2 pts) Convert 66 from octal to binary

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

   h. (2 pts) Convert 209 from decimal to base 2

   i. (2 pts) Convert 1011 0110 1110 0000 from base 2 to hexadecimal

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

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) &a

b. (2 pts) argv[0]

c. (2 pts) c->next

d. (2 pts) argc

e. (2 pts) *a

f. (2 pts) d

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

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

i. (2 pts) &w

j. (2 pts) c->data

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);
```

```
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 lime fig mango
```

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

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

c. (2 pts) What is the value of argv[2][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 011 000 011 from binary to octal

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

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

   d. (2 pts) Convert 0001 0110 1011 1100 from base 2 to hexadecimal

   e. (2 pts) Convert 17b from base 16 to base 2

   f. (2 pts) Convert 1100 from binary to decimal

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

   h. (2 pts) Convert 0011 1010 0100 1011 from base 2 to base 16

   i. (2 pts) Convert 589a from hexadecimal to base 2

   j. (2 pts) Convert 1001 1110 from binary to base 10

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) *q

b. (2 pts) argv[0]

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

d. (2 pts) g

e. (2 pts) r->next

f. (2 pts) argc

g. (2 pts) &r

h. (2 pts) s

i. (2 pts) r->data

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

k. (2 pts) &p

3. (10 pts) In 24 hour time:

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

   Assume the following struct:

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

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

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

   Comply with these instructions for full credit:

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

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

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

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

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

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

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

Comply with these instructions for full credit:

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

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

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

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

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

Comply with these instructions for full credit:

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

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

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

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

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

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

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

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

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

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

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

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

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

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

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

```
./runIt grape date cherry
```

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

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

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

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

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

1. Please perform the following number conversions.
   a. (2 pts) Convert a66d from hexadecimal to base 2
   b. (2 pts) Convert 77 from decimal to binary
   c. (2 pts) Convert 0110 1001 0101 0110 from base 2 to hexadecimal
   d. (2 pts) Convert 162 from base 10 to base 2
   e. (2 pts) Convert 71 from base 8 to binary
   f. (2 pts) Convert 101 001 100 from base 2 to base 8
   g. (2 pts) Convert 100 100 000 from binary to base 8
   h. (2 pts) Convert 60 from base 8 to base 2
   i. (2 pts) Convert 34 from base 10 to binary
   j. (2 pts) Convert 0011 1010 from base 2 to base 10

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) &e

b. (2 pts) &q

c. (2 pts) *h

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

e. (2 pts) h

f. (2 pts) g->data

g. (2 pts) f

h. (2 pts) argv[0]

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

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

```
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 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][1]?

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

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

1. Please perform the following number conversions.

   a. (2 pts) Convert fec9 from hexadecimal to base 2

   b. (2 pts) Convert 1111 0000 from base 2 to decimal

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

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

   e. (2 pts) Convert 0101 0101 from base 2 to base 10

   f. (2 pts) Convert 1010 0111 1010 0100 from base 2 to base 16

   g. (2 pts) Convert 225 from base 10 to binary

   h. (2 pts) Convert 0110 1000 from base 2 to base 10

   i. (2 pts) Convert 21 from decimal to base 2

   j. (2 pts) Convert 010 001 100 from binary to octal

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) c

b. (2 pts) &b

c. (2 pts) y

d. (2 pts) *a

e. (2 pts) argv[0]

f. (2 pts) b->next

g. (2 pts) b->data

h. (2 pts) argc

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

j. (2 pts) &w

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);
```

```
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 guava mango
```

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

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

c. (2 pts) What is the value of argv[0][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 20 from octal to base 2

   b. (2 pts) Convert 1010 1010 0110 1110 from base 2 to base 16

   c. (2 pts) Convert fc98 from base 16 to binary

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

   e. (2 pts) Convert 388b from base 16 to base 2

   f. (2 pts) Convert 4181 from base 16 to base 2

   g. (2 pts) Convert 8ba6 from hexadecimal to binary

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

   i. (2 pts) Convert df3c from base 16 to binary

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

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) y

b. (2 pts) &w

c. (2 pts) y->data

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

e. (2 pts) r

f. (2 pts) argc

g. (2 pts) argv[0]

h. (2 pts) &q

i. (2 pts) y->next

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

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);
```

```
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 kiwi cherry date
```

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

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

c. (2 pts) What is the value of argv[1][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 0111 0101 0000 1011 from base 2 to hexadecimal

   b. (2 pts) Convert 7bf4 from hexadecimal to binary

   c. (2 pts) Convert 0011 0010 from base 2 to base 10

   d. (2 pts) Convert 0100 0110 from base 2 to base 10

   e. (2 pts) Convert 1000 1110 1010 1001 from base 2 to hexadecimal

   f. (2 pts) Convert 0111 0011 1110 1010 from binary to hexadecimal

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

   h. (2 pts) Convert 0101 1001 0000 1101 from binary to base 16

   i. (2 pts) Convert 80f7 from hexadecimal to base 2

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

2. Given the following declarations:

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


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


  return 0;
}
```

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

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

b. (2 pts) e

c. (2 pts) argv[0]

d. (2 pts) argc

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

f. (2 pts) g->next

g. (2 pts) &f

h. (2 pts) *p

i. (2 pts) g->data

j. (2 pts) h

k. (2 pts) &p

3. (10 pts) In 24 hour time:

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

   Assume the following struct:

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

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

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

   Comply with these instructions for full credit:

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

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

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

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

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

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

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

Comply with these instructions for full credit:

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

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

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

Comply with these instructions for full credit:

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

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

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

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

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

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

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

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

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

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

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

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

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

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

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

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

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

```
./runIt fig mango grape
```

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

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

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

d. (2 pts) What is the value of argv[1][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 101 110 011 from binary to octal

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

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

   d. (2 pts) Convert 210 from base 10 to binary

   e. (2 pts) Convert 34 from base 8 to binary

   f. (2 pts) Convert 3 from base 8 to binary

   g. (2 pts) Convert 001 101 111 from binary to octal

   h. (2 pts) Convert 70 from base 8 to base 2

   i. (2 pts) Convert 4b0e from base 16 to binary

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

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) &c

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

c. (2 pts) *f

d. (2 pts) c->data

e. (2 pts) argc

f. (2 pts) &a

g. (2 pts) argv[0]

h. (2 pts) y

i. (2 pts) c->next

j. (2 pts) c

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 kiwi date
```

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

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

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

d. (2 pts) What is the value of argv[0][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 0001 0010 from base 2 to decimal

   b. (2 pts) Convert 1101 1000 1110 1000 from binary to hexadecimal

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

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

   e. (2 pts) Convert 226 from base 10 to base 2

   f. (2 pts) Convert 1111 from base 2 to decimal

   g. (2 pts) Convert 8978 from base 16 to base 2

   h. (2 pts) Convert 135 from decimal to base 2

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

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

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) &s

c. (2 pts) x->data

d. (2 pts) *z

e. (2 pts) x->next

f. (2 pts) argv[0]

g. (2 pts) z

h. (2 pts) q

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

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

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 grape banana
```

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

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

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

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

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

1. Please perform the following number conversions.

   a. (2 pts) Convert 5726 from hexadecimal to base 2

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

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

   d. (2 pts) Convert 0001 1001 1100 1100 from binary to base 16

   e. (2 pts) Convert 1100 0101 1011 1000 from base 2 to hexadecimal

   f. (2 pts) Convert 101 010 001 from base 2 to base 8

   g. (2 pts) Convert 3369 from hexadecimal to base 2

   h. (2 pts) Convert 1110 1101 1111 from base 2 to base 16

   i. (2 pts) Convert 7 from base 10 to binary

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

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) s

b. (2 pts) argc

c. (2 pts) h

d. (2 pts) s->data

e. (2 pts) &r

f. (2 pts) s->next

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

h. (2 pts) *q

i. (2 pts) &g

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

k. (2 pts) argv[0]

3. (10 pts) In 24 hour time:

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

   Assume the following struct:

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

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

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

   Comply with these instructions for full credit:

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

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

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

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

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

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

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

Comply with these instructions for full credit:

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

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

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

```
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 grape lemon lime
```

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][0]?

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 136 from base 10 to base 2

   b. (2 pts) Convert 0110 0100 1001 0111 from base 2 to hexadecimal

   c. (2 pts) Convert 1110 1010 0111 1000 from binary to hexadecimal

   d. (2 pts) Convert 35 from octal to base 2

   e. (2 pts) Convert 6 from base 8 to binary

   f. (2 pts) Convert 219 from decimal to binary

   g. (2 pts) Convert 3597 from base 16 to base 2

   h. (2 pts) Convert 0111 0111 1100 1110 from binary to base 16

   i. (2 pts) Convert 169 from decimal to base 2

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

2. Given the following declarations:

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


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


  return 0;
}
```

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

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

b. (2 pts) &t

c. (2 pts) r->data

d. (2 pts) &g

e. (2 pts) argc

f. (2 pts) q

g. (2 pts) s

h. (2 pts) r->next

i. (2 pts) argv[0]

j. (2 pts) *r

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

3. (10 pts) In 24 hour time:

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

   Assume the following struct:

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

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

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

   Comply with these instructions for full credit:

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

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

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

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

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

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

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

Comply with these instructions for full credit:

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

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

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

Comply with these instructions for full credit:

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

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

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

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

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

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

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

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

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

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

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

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

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

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

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

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

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

```
./runIt grape banana 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[2][5]?

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 1100 1101 0110 0111 from base 2 to hexadecimal

   b. (2 pts) Convert 0001 1110 from binary to base 10

   c. (2 pts) Convert 216 from decimal to binary

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

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

   f. (2 pts) Convert 117 from decimal to binary

   g. (2 pts) Convert df88 from hexadecimal to base 2

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

   i. (2 pts) Convert 736a from base 16 to binary

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

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) z

b. (2 pts) a->next

c. (2 pts) argv[0]

d. (2 pts) &y

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

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

g. (2 pts) *a

h. (2 pts) &a

i. (2 pts) a->data

j. (2 pts) argc

k. (2 pts) x

3. (10 pts) In 24 hour time:

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

   Assume the following struct:

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

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

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

   Comply with these instructions for full credit:

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

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

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

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

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

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

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

Comply with these instructions for full credit:

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

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

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

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

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

Comply with these instructions for full credit:

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

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

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

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

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

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

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

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

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

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

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

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

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

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

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

```
./runIt fig cherry
```

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][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 226 from base 10 to base 2

   b. (2 pts) Convert 1101 1101 1101 1100 from base 2 to hexadecimal

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

   d. (2 pts) Convert 80 from decimal to binary

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

   f. (2 pts) Convert d542 from base 16 to base 2

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

   h. (2 pts) Convert 235 from base 10 to binary

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

   j. (2 pts) Convert 101 110 000 from base 2 to octal

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) x

b. (2 pts) q

c. (2 pts) *t

d. (2 pts) &w

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

f. (2 pts) &p

g. (2 pts) t->data

h. (2 pts) argc

i. (2 pts) t->next

j. (2 pts) argv[0]

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

3. (10 pts) In 24 hour time:

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

   Assume the following struct:

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

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

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

   Comply with these instructions for full credit:

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

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

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

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

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

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

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

Comply with these instructions for full credit:

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

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

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

```
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 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][2]?

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

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

1. Please perform the following number conversions.

   a. (2 pts) Convert 0010 0111 0100 0001 from base 2 to base 16

   b. (2 pts) Convert 1001 1000 from base 2 to base 10

   c. (2 pts) Convert 0100 0111 0100 0110 from binary to hexadecimal

   d. (2 pts) Convert 220 from decimal to base 2

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

   f. (2 pts) Convert 111 from base 10 to base 2

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

   h. (2 pts) Convert 0111 0011 0101 0000 from binary to hexadecimal

   i. (2 pts) Convert 0101 1011 from base 2 to base 10

   j. (2 pts) Convert 25 from octal to binary

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) r

b. (2 pts) e

c. (2 pts) *r

d. (2 pts) q->next

e. (2 pts) argc

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

g. (2 pts) &p

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

i. (2 pts) q->data

j. (2 pts) argv[0]

k. (2 pts) &f

3. (10 pts) In 24 hour time:

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

   Assume the following struct:

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

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

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

   Comply with these instructions for full credit:

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

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

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

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

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

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

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

Comply with these instructions for full credit:

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

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

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

```
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 mango
```

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][2]?

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 58a3 from hexadecimal to binary

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

   c. (2 pts) Convert 0111 1101 from binary to decimal

   d. (2 pts) Convert 38cb from hexadecimal to binary

   e. (2 pts) Convert 1100 0111 from base 2 to base 10

   f. (2 pts) Convert 50 from octal to binary

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

   h. (2 pts) Convert 1101 1100 0011 1111 from binary to hexadecimal

   i. (2 pts) Convert 1111 1101 from binary to decimal

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

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) d->data

b. (2 pts) argc

c. (2 pts) c

d. (2 pts) d->next

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

f. (2 pts) y

g. (2 pts) argv[0]

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

i. (2 pts) &a

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 apple kiwi lemon fig
```

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][2]?

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

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

1. Please perform the following number conversions.
   a. (2 pts) Convert 47 from octal to binary

   b. (2 pts) Convert 23b3 from hexadecimal to base 2

   c. (2 pts) Convert 32 from octal to binary

   d. (2 pts) Convert 1100 0100 1011 0101 from binary to base 16

   e. (2 pts) Convert aa5f from base 16 to binary

   f. (2 pts) Convert 0011 1011 0110 0101 from binary to hexadecimal

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

   h. (2 pts) Convert 0110 0011 from base 2 to decimal

   i. (2 pts) Convert 199 from base 10 to base 2

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

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) &w

b. (2 pts) z->next

c. (2 pts) argc

d. (2 pts) argv[0]

e. (2 pts) z->data

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

g. (2 pts) &y

h. (2 pts) y

i. (2 pts) s

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 cherry date apple
```

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

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

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

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 111 101 011 from binary to octal

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

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

   d. (2 pts) Convert 1000 0000 from binary to base 10

   e. (2 pts) Convert 0001 1010 from base 2 to base 10

   f. (2 pts) Convert 60 from decimal to binary

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

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

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

   j. (2 pts) Convert 11 from octal to base 2

2. Given the following declarations:

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


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


  return 0;
}
```

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

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

b. (2 pts) p->data

c. (2 pts) &s

d. (2 pts) argv[0]

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

f. (2 pts) *s

g. (2 pts) argc

h. (2 pts) h

i. (2 pts) &g

j. (2 pts) p->next

k. (2 pts) s

3. (10 pts) In 24 hour time:

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

   Assume the following struct:

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

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

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

   Comply with these instructions for full credit:

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

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

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

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

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

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

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

Comply with these instructions for full credit:

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

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

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

Comply with these instructions for full credit:

```
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[0][0]?

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

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 58 from base 10 to base 2

   b. (2 pts) Convert 128 from base 10 to base 2

   c. (2 pts) Convert 1111 1110 from binary to base 10

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

   e. (2 pts) Convert fe23 from base 16 to base 2

   f. (2 pts) Convert 1101 0110 from base 2 to base 10

   g. (2 pts) Convert 208 from base 10 to binary

   h. (2 pts) Convert 1001 0010 0001 0001 from base 2 to hexadecimal

   i. (2 pts) Convert 100 001 000 from binary to octal

   j. (2 pts) Convert 60 from base 8 to base 2

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) f->next

b. (2 pts) *e

c. (2 pts) &b

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

e. (2 pts) &f

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

g. (2 pts) f->data

h. (2 pts) argc

i. (2 pts) g

j. (2 pts) b

k. (2 pts) argv[0]

3. (10 pts) In 24 hour time:

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

   Assume the following struct:

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

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

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

   Comply with these instructions for full credit:

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

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

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

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

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

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

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

Comply with these instructions for full credit:

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

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

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

```
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 lime
```

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][0]?

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

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

1. Please perform the following number conversions.

   a. (2 pts) Convert 0110 1100 from base 2 to decimal

   b. (2 pts) Convert 0101 0010 from base 2 to decimal

   c. (2 pts) Convert 0011 0101 0010 0110 from binary to base 16

   d. (2 pts) Convert 0110 1000 from binary to decimal

   e. (2 pts) Convert 84 from base 10 to binary

   f. (2 pts) Convert 8fb from base 16 to base 2

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

   h. (2 pts) Convert 76 from octal to binary

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

   j. (2 pts) Convert 74 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;
  int s;
  Node t;
  char w;
  double *x;
  int *y;
  Node *z;
  char *a;


  return 0;
}
```

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

a. (2 pts) y

b. (2 pts) *x

c. (2 pts) &r

d. (2 pts) argv[0]

e. (2 pts) w

f. (2 pts) &z

g. (2 pts) z->next

h. (2 pts) z->data

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

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

k. (2 pts) argc

3. (10 pts) In 24 hour time:

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

   Assume the following struct:

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

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

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

   Comply with these instructions for full credit:

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

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

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

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

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

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

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

Comply with these instructions for full credit:

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

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

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

```
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 guava lemon lime
```

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][0]?

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 b0ff from hexadecimal to binary
   b. (2 pts) Convert 1100 0101 0110 from binary to base 16
   c. (2 pts) Convert 35 from base 10 to base 2
   d. (2 pts) Convert 1111 0100 from binary to decimal
   e. (2 pts) Convert 001 101 111 from base 2 to octal
   f. (2 pts) Convert a2d8 from hexadecimal to binary
   g. (2 pts) Convert 011 111 001 from base 2 to octal
   h. (2 pts) Convert 1000 0010 from binary to decimal
   i. (2 pts) Convert 1110 1111 from binary to base 10
   j. (2 pts) Convert 42 from octal to base 2

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) s

b. (2 pts) argv[0]

c. (2 pts) &r

d. (2 pts) argc

e. (2 pts) p

f. (2 pts) t->next

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

h. (2 pts) &s

i. (2 pts) *t

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

k. (2 pts) t->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 lime date
```

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][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 95c from hexadecimal to binary
   b. (2 pts) Convert af22 from hexadecimal to binary
   c. (2 pts) Convert c869 from hexadecimal to binary
   d. (2 pts) Convert 53 from base 8 to base 2
   e. (2 pts) Convert 168 from base 10 to base 2
   f. (2 pts) Convert 1010 0100 0010 1001 from binary to hexadecimal
   g. (2 pts) Convert ce35 from hexadecimal to base 2
   h. (2 pts) Convert 0010 1001 0010 0010 from binary to base 16
   i. (2 pts) Convert 1110 0010 from base 2 to base 10
   j. (2 pts) Convert 151 from base 10 to binary

2. Given the following declarations:

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


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


  return 0;
}
```

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

a. (2 pts) e->data

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

c. (2 pts) a

d. (2 pts) *f

e. (2 pts) argv[0]

f. (2 pts) g

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

h. (2 pts) &c

i. (2 pts) e->next

j. (2 pts) &h

k. (2 pts) argc

3. (10 pts) In 24 hour time:

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

   Assume the following struct:

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

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

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

   Comply with these instructions for full credit:

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

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

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt kiwi banana
```

a. (2 pts) What is the value of `argc` in this case?

b. (2 pts) What is the value of argv[2][1]?

c. (2 pts) What is the value of argv[1][1]?

d. (2 pts) What is the value of argv[0][5]?

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 010 011 100 from binary to octal

   b. (2 pts) Convert 0110 1001 0100 1011 from binary to hexadecimal

   c. (2 pts) Convert 182 from base 10 to binary

   d. (2 pts) Convert 3bdb from base 16 to binary

   e. (2 pts) Convert 1000 1011 0101 0000 from base 2 to hexadecimal

   f. (2 pts) Convert 3e06 from hexadecimal to binary

   g. (2 pts) Convert 7826 from base 16 to base 2

   h. (2 pts) Convert 54 from base 8 to base 2

   i. (2 pts) Convert 101 011 000 from binary to octal

   j. (2 pts) Convert 51 from decimal to binary

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int x;
  double y;
  Node z;
  char a;
  int *b;
  double *c;
  Node *d;
  char *e;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) &d

b. (2 pts) &z

c. (2 pts) d->next->next

d. (2 pts) argc

e. (2 pts) d->data

f. (2 pts) c

g. (2 pts) a

h. (2 pts) argv[1][2]

i. (2 pts) argv[0]

j. (2 pts) *d

k. (2 pts) d->next

3. (10 pts) In 24 hour time:

   - midnight is represented by the hour 0
   - 1am through 11am are represented at hour=1 through hour=11
   - noon is represented by hour=12
   - 1pm through 11pm are represetned by hour=13 through hour=23
   - minute is a value between 0 and 59.

   Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

   Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

   Comply with these instructions for full credit:

   - The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
   - Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Comply with these instructions for full credit:

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt apple grape fig cherry
```

a. (2 pts) What is the value of `argc` in this case?

b. (2 pts) What is the value of argv[1][2]?

c. (2 pts) What is the value of argv[0][6]?

d. (2 pts) What is the value of argv[2][3]?

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 0111 1111 1001 1101 from base 2 to base 16

   b. (2 pts) Convert 16 from base 8 to binary

   c. (2 pts) Convert 73 from base 8 to base 2

   d. (2 pts) Convert 100 110 000 from base 2 to base 8

   e. (2 pts) Convert 70 from base 8 to binary

   f. (2 pts) Convert 011 100 000 from base 2 to base 8

   g. (2 pts) Convert 122 from decimal to base 2

   h. (2 pts) Convert 6 from base 8 to binary

   i. (2 pts) Convert 0100 1110 from base 2 to base 10

   j. (2 pts) Convert 27 from octal to base 2

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  Node x;
  int y;
  double z;
  char a;
  Node *b;
  int *c;
  double *d;
  char *e;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) argv[1][2]

b. (2 pts) *d

c. (2 pts) b->next->next

d. (2 pts) x

e. (2 pts) &e

f. (2 pts) argv[0]

g. (2 pts) argc

h. (2 pts) b

i. (2 pts) b->next

j. (2 pts) &x

k. (2 pts) b->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 cherry grape 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[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 1100 0100 from base 2 to decimal
   b. (2 pts) Convert 111 101 001 from binary to base 8
   c. (2 pts) Convert da88 from hexadecimal to binary
   d. (2 pts) Convert 0010 0100 0000 0110 from base 2 to hexadecimal
   e. (2 pts) Convert 110 001 001 from base 2 to base 8
   f. (2 pts) Convert 2 from octal to binary
   g. (2 pts) Convert 2445 from base 16 to base 2
   h. (2 pts) Convert 1010 0001 from binary to decimal
   i. (2 pts) Convert 110 000 from binary to octal
   j. (2 pts) Convert 251 from decimal to binary

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double d;
  Node e;
  int f;
  char g;
  double *h;
  Node *p;
  int *q;
  char *r;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) &p

b. (2 pts) p->next->next

c. (2 pts) argc

d. (2 pts) *r

e. (2 pts) p->data

f. (2 pts) argv[1][2]

g. (2 pts) p->next

h. (2 pts) argv[0]

i. (2 pts) f

j. (2 pts) r

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 cherry
```

a. (2 pts) What is the value of `argc` in this case?

b. (2 pts) What is the value of argv[2][0]?

c. (2 pts) What is the value of argv[0][0]?

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 0001 1100 from base 2 to base 10

   b. (2 pts) Convert 1001 0111 1100 0101 from binary to base 16

   c. (2 pts) Convert 0111 1111 1110 1010 from base 2 to base 16

   d. (2 pts) Convert 223 from base 10 to binary

   e. (2 pts) Convert 3532 from base 16 to binary

   f. (2 pts) Convert b9c from hexadecimal to binary

   g. (2 pts) Convert 35 from octal to binary

   h. (2 pts) Convert 0100 0111 1110 0100 from binary to base 16

   i. (2 pts) Convert 010 101 from base 2 to base 8

   j. (2 pts) Convert 0111 from binary to decimal

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double e;
  Node f;
  int g;
  char h;
  double *p;
  Node *q;
  int *r;
  char *s;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) q->data

b. (2 pts) f

c. (2 pts) r

d. (2 pts) argv[0]

e. (2 pts) q->next

f. (2 pts) &p

g. (2 pts) *r

h. (2 pts) argc

i. (2 pts) &h

j. (2 pts) q->next->next

k. (2 pts) argv[1][2]

3. (10 pts) In 24 hour time:

   - midnight is represented by the hour 0
   - 1am through 11am are represented at hour=1 through hour=11
   - noon is represented by hour=12
   - 1pm through 11pm are represetned by hour=13 through hour=23
   - minute is a value between 0 and 59.

   Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

   Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

   Comply with these instructions for full credit:

   - The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
   - Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

```
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 apple
```

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][1]?

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 0110 0001 1011 1000 from base 2 to hexadecimal

   b. (2 pts) Convert 0101 0001 from binary to decimal

   c. (2 pts) Convert 0110 1101 1100 1011 from binary to base 16

   d. (2 pts) Convert 107 from base 10 to base 2

   e. (2 pts) Convert 0001 1000 0111 1110 from base 2 to hexadecimal

   f. (2 pts) Convert 165 from decimal to binary

   g. (2 pts) Convert 7 from base 8 to binary

   h. (2 pts) Convert 63 from octal to base 2

   i. (2 pts) Convert 65 from octal to binary

   j. (2 pts) Convert 1010 0011 from base 2 to base 10

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double r;
  int s;
  Node t;
  char w;
  double *x;
  int *y;
  Node *z;
  char *a;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) &z

b. (2 pts) r

c. (2 pts) z->next

d. (2 pts) argc

e. (2 pts) z->data

f. (2 pts) *x

g. (2 pts) argv[0]

h. (2 pts) &t

i. (2 pts) y

j. (2 pts) argv[1][2]

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);
```

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 date guava grape 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[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 147 from decimal to base 2

   b. (2 pts) Convert 001 000 110 from base 2 to base 8

   c. (2 pts) Convert 164 from decimal to binary

   d. (2 pts) Convert c7c5 from base 16 to binary

   e. (2 pts) Convert 33 from base 8 to base 2

   f. (2 pts) Convert 65 from base 8 to base 2

   g. (2 pts) Convert 2216 from hexadecimal to binary

   h. (2 pts) Convert 0011 1000 from base 2 to base 10

   i. (2 pts) Convert 0111 0110 from base 2 to base 10

   j. (2 pts) Convert 207 from base 10 to binary

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int s;
  Node t;
  double w;
  char x;
  int *y;
  Node *z;
  double *a;
  char *b;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) &y

b. (2 pts) z->next

c. (2 pts) z->next->next

d. (2 pts) y

e. (2 pts) argv[0]

f. (2 pts) &t

g. (2 pts) argv[1][2]

h. (2 pts) z->data

i. (2 pts) argc

j. (2 pts) *y

k. (2 pts) t

3. (10 pts) In 24 hour time:

   - midnight is represented by the hour 0
   - 1am through 11am are represented at hour=1 through hour=11
   - noon is represented by hour=12
   - 1pm through 11pm are represetned by hour=13 through hour=23
   - minute is a value between 0 and 59.

   Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

   Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

   Comply with these instructions for full credit:

   - The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
   - Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt apple mango fig
```

a. (2 pts) What is the value of `argc` in this case?

b. (2 pts) What is the value of argv[0][0]?

c. (2 pts) What is the value of argv[1][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 65 from base 8 to base 2

   b. (2 pts) Convert 1101 1101 from base 2 to decimal

   c. (2 pts) Convert 100 100 100 from binary to octal

   d. (2 pts) Convert 0101 0011 1010 1111 from base 2 to base 16

   e. (2 pts) Convert 0101 0001 from base 2 to base 10

   f. (2 pts) Convert 0111 0001 1100 0000 from base 2 to base 16

   g. (2 pts) Convert cc07 from base 16 to binary

   h. (2 pts) Convert 192 from base 10 to base 2

   i. (2 pts) Convert 20 from octal to binary

   j. (2 pts) Convert 107 from base 10 to binary

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double h;
  Node p;
  int q;
  char r;
  double *s;
  Node *t;
  int *w;
  char *x;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) argc

b. (2 pts) t->data

c. (2 pts) x

d. (2 pts) &h

e. (2 pts) *t

f. (2 pts) t->next

g. (2 pts) argv[1][2]

h. (2 pts) r

i. (2 pts) t->next->next

j. (2 pts) argv[0]

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);
```

```
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 mango
```

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 45 from base 8 to binary
   b. (2 pts) Convert 59 from decimal to binary
   c. (2 pts) Convert 0111 1011 0110 0000 from base 2 to base 16
   d. (2 pts) Convert 44 from base 8 to binary
   e. (2 pts) Convert 010 110 001 from base 2 to octal
   f. (2 pts) Convert 0011 0111 from binary to decimal
   g. (2 pts) Convert dfa6 from base 16 to binary
   h. (2 pts) Convert 34 from base 8 to binary
   i. (2 pts) Convert 1111 0000 from base 2 to decimal
   j. (2 pts) Convert f995 from hexadecimal to base 2

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

b. (2 pts) c->data

c. (2 pts) argv[0]

d. (2 pts) &a

e. (2 pts) argc

f. (2 pts) b

g. (2 pts) argv[1][2]

h. (2 pts) &e

i. (2 pts) c->next

j. (2 pts) c->next->next

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);
```

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Comply with these instructions for full credit:

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt lime lemon
```

a. (2 pts) What is the value of `argc` in this case?

b. (2 pts) What is the value of argv[2][3]?

c. (2 pts) What is the value of argv[1][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 1101 1001 from binary to base 10

   b. (2 pts) Convert 75 from octal to base 2

   c. (2 pts) Convert 6940 from base 16 to base 2

   d. (2 pts) Convert 7 from base 8 to base 2

   e. (2 pts) Convert 0011 1100 from base 2 to base 10

   f. (2 pts) Convert 110 100 010 from binary to octal

   g. (2 pts) Convert 100 010 011 from binary to base 8

   h. (2 pts) Convert 1111 1000 from base 2 to decimal

   i. (2 pts) Convert 186 from decimal to base 2

   j. (2 pts) Convert 149 from decimal to base 2

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int s;
  double t;
  Node w;
  char x;
  int *y;
  double *z;
  Node *a;
  char *b;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) a->next->next

b. (2 pts) argc

c. (2 pts) a->next

d. (2 pts) &z

e. (2 pts) b

f. (2 pts) a->data

g. (2 pts) argv[0]

h. (2 pts) s

i. (2 pts) argv[1][2]

j. (2 pts) &w

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 cherry guava fig 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][1]?

d. (2 pts) What is the value of argv[0][1]?

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 010 101 from base 2 to octal

   b. (2 pts) Convert 1100 0110 1111 1110 from base 2 to hexadecimal

   c. (2 pts) Convert 100 111 111 from base 2 to base 8

   d. (2 pts) Convert 011 110 100 from base 2 to octal

   e. (2 pts) Convert 146 from base 10 to base 2

   f. (2 pts) Convert 3b9 from base 16 to base 2

   g. (2 pts) Convert 8bc5 from base 16 to binary

   h. (2 pts) Convert 97 from decimal to base 2

   i. (2 pts) Convert 92 from base 10 to base 2

   j. (2 pts) Convert c1c0 from hexadecimal to base 2

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double e;
  int f;
  Node g;
  char h;
  double *p;
  int *q;
  Node *r;
  char *s;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) s

b. (2 pts) r->next->next

c. (2 pts) *r

d. (2 pts) argv[1][2]

e. (2 pts) argv[0]

f. (2 pts) f

g. (2 pts) &s

h. (2 pts) r->next

i. (2 pts) &f

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 grape date apple 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][5]?

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 79 from base 10 to binary
   b. (2 pts) Convert 40 from base 8 to base 2
   c. (2 pts) Convert 100 011 010 from binary to base 8
   d. (2 pts) Convert 0110 0001 0010 from base 2 to base 16
   e. (2 pts) Convert 0111 0101 1010 1001 from base 2 to hexadecimal
   f. (2 pts) Convert 157 from base 10 to binary
   g. (2 pts) Convert 35b6 from hexadecimal to binary
   h. (2 pts) Convert 232 from base 10 to base 2
   i. (2 pts) Convert 261b from hexadecimal to base 2
   j. (2 pts) Convert 5daa from base 16 to base 2

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  Node b;
  int c;
  double d;
  char e;
  Node *f;
  int *g;
  double *h;
  char *p;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) argc

b. (2 pts) e

c. (2 pts) *f

d. (2 pts) &b

e. (2 pts) &g

f. (2 pts) f->next->next

g. (2 pts) argv[0]

h. (2 pts) argv[1][2]

i. (2 pts) h

j. (2 pts) f->data

k. (2 pts) f->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 grape banana
```

a. (2 pts) What is the value of `argc` in this case?

b. (2 pts) What is the value of argv[2][2]?

c. (2 pts) What is the value of argv[0][0]?

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 167 from base 10 to binary
   b. (2 pts) Convert 23f3 from base 16 to base 2
   c. (2 pts) Convert 14 from base 8 to binary
   d. (2 pts) Convert 193 from base 10 to binary
   e. (2 pts) Convert 230 from base 10 to base 2
   f. (2 pts) Convert 47 from base 8 to binary
   g. (2 pts) Convert 100 001 110 from base 2 to octal
   h. (2 pts) Convert 43 from octal to binary
   i. (2 pts) Convert 188f from hexadecimal to binary
   j. (2 pts) Convert 105 from base 10 to binary

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double s;
  Node t;
  int w;
  char x;
  double *y;
  Node *z;
  int *a;
  char *b;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) y

b. (2 pts) argv[1][2]

c. (2 pts) z->next

d. (2 pts) x

e. (2 pts) z->next->next

f. (2 pts) *b

g. (2 pts) &y

h. (2 pts) argv[0]

i. (2 pts) z->data

j. (2 pts) argc

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 date 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[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 73 from base 8 to binary
   b. (2 pts) Convert 110 111 100 from binary to octal
   c. (2 pts) Convert 10 from octal to binary
   d. (2 pts) Convert 77 from base 10 to binary
   e. (2 pts) Convert 1100 1001 0110 1101 from base 2 to hexadecimal
   f. (2 pts) Convert 16 from octal to binary
   g. (2 pts) Convert 001 100 010 from binary to octal
   h. (2 pts) Convert 0001 0111 from base 2 to decimal
   i. (2 pts) Convert 1110 0010 from base 2 to base 10
   j. (2 pts) Convert 5 from base 10 to binary

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double h;
  int p;
  Node q;
  char r;
  double *s;
  int *t;
  Node *w;
  char *x;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) x

b. (2 pts) w->data

c. (2 pts) w->next

d. (2 pts) p

e. (2 pts) argv[1][2]

f. (2 pts) w->next->next

g. (2 pts) argc

h. (2 pts) &r

i. (2 pts) &w

j. (2 pts) *s

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 date banana fig kiwi
```

a. (2 pts) What is the value of `argc` in this case?

b. (2 pts) What is the value of argv[1][0]?

c. (2 pts) What is the value of argv[2][0]?

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 1e2b from hexadecimal to base 2
   b. (2 pts) Convert 53 from octal to base 2
   c. (2 pts) Convert 5720 from hexadecimal to base 2
   d. (2 pts) Convert a9d0 from base 16 to binary
   e. (2 pts) Convert 1f8b from base 16 to binary
   f. (2 pts) Convert 0110 1011 from binary to base 10
   g. (2 pts) Convert 001 100 111 from base 2 to base 8
   h. (2 pts) Convert 127 from decimal to binary
   i. (2 pts) Convert 1000 0100 from base 2 to decimal
   j. (2 pts) Convert 49 from base 10 to base 2

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int p;
  double q;
  Node r;
  char s;
  int *t;
  double *w;
  Node *x;
  char *y;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) x->data

b. (2 pts) &r

c. (2 pts) r

d. (2 pts) x->next->next

e. (2 pts) *t

f. (2 pts) x->next

g. (2 pts) &t

h. (2 pts) argc

i. (2 pts) argv[0]

j. (2 pts) x

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 cherry fig lemon 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][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 011 000 110 from binary to octal

   b. (2 pts) Convert 105 from base 10 to binary

   c. (2 pts) Convert 0100 0101 from binary to decimal

   d. (2 pts) Convert 0011 0101 1011 1011 from base 2 to base 16

   e. (2 pts) Convert 0 from base 8 to binary

   f. (2 pts) Convert 001 010 from base 2 to base 8

   g. (2 pts) Convert 110 111 010 from binary to base 8

   h. (2 pts) Convert 0111 1010 1011 from binary to hexadecimal

   i. (2 pts) Convert 78 from base 10 to base 2

   j. (2 pts) Convert 205 from decimal to binary

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int x;
  Node y;
  double z;
  char a;
  int *b;
  Node *c;
  double *d;
  char *e;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) argv[1][2]

b. (2 pts) &y

c. (2 pts) x

d. (2 pts) c->next

e. (2 pts) *c

f. (2 pts) c->data

g. (2 pts) argc

h. (2 pts) c

i. (2 pts) argv[0]

j. (2 pts) &d

k. (2 pts) c->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 cherry guava fig
```

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][5]?

d. (2 pts) What is the value of argv[1][3]?

End of Exam

total points=100

CS16—Final Exam

E03, F14, Phill Conrad, UC Santa Barbara

Wednesday, 12/15/2014

Name: _____

Umail Address: _____@ umail.ucsb.edu

- Please write your name **above AND AT THE TOP OF EVERY PAGE**
- Be sure you turn in every page of this exam.
  - Each exam is numbered (e.g. Exam #137).
  - Each pages is numbered (e.g. Page 1, Page 2, etc.)
  - The last page clearly says "End of Exam".
- This exam is **closed book, closed notes**, **closed mouth, cell phone off**
- You are permitted **one sheet of paper** (max size 8.5x11") on which to write notes
- These sheets will be collected with the exam, and might not be returned
- Please write your name on your notes sheet

1. Please perform the following number conversions.

   a. (2 pts) Convert 101 110 110 from binary to octal

   b. (2 pts) Convert 1100 from base 2 to base 10

   c. (2 pts) Convert 234 from decimal to base 2

   d. (2 pts) Convert 1111 0001 from base 2 to base 10

   e. (2 pts) Convert 734f from base 16 to base 2

   f. (2 pts) Convert 6 from base 10 to base 2

   g. (2 pts) Convert 2f2b from base 16 to base 2

   h. (2 pts) Convert 1010 1110 from binary to decimal

   i. (2 pts) Convert 64 from decimal to binary

   j. (2 pts) Convert 1101 1000 from binary to base 10

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  Node h;
  double p;
  int q;
  char r;
  Node *s;
  double *t;
  int *w;
  char *x;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) s->next->next

b. (2 pts) *t

c. (2 pts) &t

d. (2 pts) s

e. (2 pts) argc

f. (2 pts) s->data

g. (2 pts) &h

h. (2 pts) h

i. (2 pts) s->next

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);
```

```
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[0][5]?

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 46 from hexadecimal to binary

   b. (2 pts) Convert 198 from decimal to binary

   c. (2 pts) Convert 110 110 000 from binary to base 8

   d. (2 pts) Convert 011 111 010 from base 2 to octal

   e. (2 pts) Convert 25 from base 8 to binary

   f. (2 pts) Convert 1010 0000 from base 2 to decimal

   g. (2 pts) Convert d91c from hexadecimal to base 2

   h. (2 pts) Convert 53 from decimal to binary

   i. (2 pts) Convert 1011 from base 2 to base 10

   j. (2 pts) Convert 0111 0100 from binary to base 10

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  Node e;
  double f;
  int g;
  char h;
  Node *p;
  double *q;
  int *r;
  char *s;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) &h

b. (2 pts) g

c. (2 pts) argv[0]

d. (2 pts) p->next

e. (2 pts) &s

f. (2 pts) s

g. (2 pts) *r

h. (2 pts) argc

i. (2 pts) argv[1][2]

j. (2 pts) p->next->next

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 kiwi 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][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 0001 from base 2 to base 10

   b. (2 pts) Convert 1001 1000 from binary to base 10

   c. (2 pts) Convert 1110 1010 0010 from binary to hexadecimal

   d. (2 pts) Convert 1101 1001 from base 2 to base 10

   e. (2 pts) Convert 101 011 001 from base 2 to octal

   f. (2 pts) Convert 1101 0010 1010 0001 from binary to hexadecimal

   g. (2 pts) Convert 110 110 110 from base 2 to octal

   h. (2 pts) Convert 158 from base 10 to binary

   i. (2 pts) Convert 172 from decimal to binary

   j. (2 pts) Convert 1010 0001 from binary to decimal

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double e;
  int f;
  Node g;
  char h;
  double *p;
  int *q;
  Node *r;
  char *s;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) r

b. (2 pts) *r

c. (2 pts) &p

d. (2 pts) h

e. (2 pts) r->next->next

f. (2 pts) argc

g. (2 pts) r->next

h. (2 pts) argv[1][2]

i. (2 pts) &g

j. (2 pts) argv[0]

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 date cherry lemon mango
```

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][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 0110 1000 1000 from binary to base 16

   b. (2 pts) Convert 0101 0010 0110 1101 from base 2 to hexadecimal

   c. (2 pts) Convert 252 from base 10 to binary

   d. (2 pts) Convert 0110 0101 from base 2 to decimal

   e. (2 pts) Convert 1001 0000 from base 2 to base 10

   f. (2 pts) Convert 6c7e from base 16 to base 2

   g. (2 pts) Convert 100 001 010 from binary to octal

   h. (2 pts) Convert 11 from octal to binary

   i. (2 pts) Convert 0111 0110 from base 2 to base 10

   j. (2 pts) Convert 0011 1100 from base 2 to decimal

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  int b;
  Node c;
  double d;
  char e;
  int *f;
  Node *g;
  double *h;
  char *p;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) g->next

b. (2 pts) &c

c. (2 pts) g->next->next

d. (2 pts) g->data

e. (2 pts) argv[0]

f. (2 pts) c

g. (2 pts) argc

h. (2 pts) *p

i. (2 pts) g

j. (2 pts) argv[1][2]

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 apple fig grape
```

a. (2 pts) What is the value of `argc` in this case?

b. (2 pts) What is the value of argv[2][0]?

c. (2 pts) What is the value of argv[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 1100 1110 1110 0100 from base 2 to base 16
   b. (2 pts) Convert 245 from base 10 to base 2
   c. (2 pts) Convert a1e4 from hexadecimal to base 2
   d. (2 pts) Convert 10 from octal to binary
   e. (2 pts) Convert 0111 1101 from base 2 to hexadecimal
   f. (2 pts) Convert 0110 1101 from base 2 to base 10
   g. (2 pts) Convert d6ee from base 16 to binary
   h. (2 pts) Convert 1100 1100 from binary to decimal
   i. (2 pts) Convert 0110 1001 from binary to decimal
   j. (2 pts) Convert 010 010 001 from binary to base 8

2. Given the following declarations:

```
struct Node {
  int data;
  Node *next;
};


int main(int argc, char *argv[]) {
  double s;
  Node t;
  int w;
  char x;
  double *y;
  Node *z;
  int *a;
  char *b;


  return 0;
}
```

Specify the type of each of these expressions (e.g. `int`, `int *`, etc.

a. (2 pts) *b

b. (2 pts) t

c. (2 pts) argv[1][2]

d. (2 pts) z->data

e. (2 pts) b

f. (2 pts) argc

g. (2 pts) argv[0]

h. (2 pts) z->next->next

i. (2 pts) &z

j. (2 pts) &t

k. (2 pts) z->next

3. (10 pts) In 24 hour time:

   - midnight is represented by the hour 0
   - 1am through 11am are represented at hour=1 through hour=11
   - noon is represented by hour=12
   - 1pm through 11pm are represetned by hour=13 through hour=23
   - minute is a value between 0 and 59.

   Assume the following struct:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

   Write the full function definition for a function that would have the following prototype:

```
  bool isValidTime(Time t);
```

   Comply with these instructions for full credit:

   - The function should return true if both of the members of the struct t are valid values for a struct Time. Otherwise the function should return false.
   - Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

4. (10 pts) Given the same struct definition as in the previous problem, with the same understanding of how are the fields for hour and minutes are to be treated in 24 hour time:

```
// Time of day in 24 hour time
struct Time {
   int h; // hour
   int m; // minute
};
```

And given the same function prototype for `isValidTime`, which you can assume is ALREADY DEFINED and available for you to use (do not redefine it in this problem)

```
bool isValidTime(Time t);
```

Write the full function definition for a function that would have the following prototype:

```
bool setTime(Time *t, int hour, int min);
```

Comply with these instructions for full credit:

- If the pointer t is NULL, return false, and do nothing.
- If the pointer t is not NULL, copy the values from hour and min into the corresponding fields in the struct pointed to by t.
- Then, USING A FUNCTION CALL TO isValidTime, NOT BY REPEATED THE CODE YOU WROTE FOR THAT FUNCTION, check whether the struct you just initialized is valid. If so, return true, otherwise return false. Hint: be careful about matching the types of formal and actual parameters.

5. (10 pts) Given the struct definitions shown below (which are exactly as in lab09 and lab09), write the full function definition for a function that would have the following prototype:

```
int countOccurences(LinkedList * list, int k);
```

Comply with these instructions for full credit:

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

- The function should return 0 if `list` is NULL, or the the linked list that is pointed to is an empty list.
- Otherwise, the function should go through the linked list and count the number of nodes in the list that have the value `k` in the data member.
- Assume that any `#include` directives you need are ALREADY DONE. I ONLY want the function definition. I want the full function definition, and NOTHING ELSE. PERIOD.

6. (10 pts) Again, given the following struct definitions, exactly as in lab09 and lab09

```
struct Node {
  int data;
  Node *next;
};

struct LinkedList {
  Node *head;
  Node *tail;
};
```

Here is a partial implementation of addIntToEndOfList. It has one extra line of code added that will result in a memory leak, and it has one crucial line of code missing.

You have two jobs, and you must do this precisely as stated to get full credit:

- (5 pts) Add EXACTLY ONE LINE OF CODE, the missing line that will cause the function to work properly.
- (5 pts) Cross out EXACTLY ONE LINE OF CODE, the one that causes a memory leak.

HINT: Trace through the code with a picture, both starting from an empty list, and one that already has at least one node in it. Think about what you WANT the "after" picture to look like, vs. what the code you see here actualy does. Then you'll see both thing you need to fix.

NOTE: there may be 'other' ways to fix the code. I DON'T WANT THOSE. Fix it according to the instructions. Also, note that I've put exactly the same amount of space between each line, so that where the white space is is NOT a clue as to where the missing line of code should go.

```
void addIntToEndOfList(LinkedList *list, int value) {

 assert(list!=NULL); // if list is NULL, we can do nothing.

  Node *p;

  p = new Node;

  p->data = value;

  p->next = NULL;

  if (list->head == NULL) {

     list->head = new Node;

     list->head = p;

  } else {

     list->tail->next = p;

     list -> tail = p;

  }

}
```

7. Assume the `main` function in the program `runIt.cpp` starts with:

```
int main(int argc, char *argv[]) {
...
```

Further, suppose this program is invoked with the following command line:

```
./runIt banana lemon
```

a. (2 pts) What is the value of `argc` in this case?

b. (2 pts) What is the value of argv[1][5]?

c. (2 pts) What is the value of argv[2][1]?

d. (2 pts) What is the value of argv[0][4]?

End of Exam

total points=100