CGS 3460 - Programming Using C
Summer 2009

Assignment #8
20 Points
Assigned: 07/17/2009, Friday
Due: 07/23/2009, Thu, 11:59 PM

At this point in the semester we have discussed how to use header files. Most, if not all of the remaining homework problems will require you to declare and define functions using a pX.c and a pX.h file. The pX.h file will then be included from our file which contains the main() function and your functions will be tested from our main() function. Please be sure to pay close attention to the name of the files specified.

1. **(5 Points)** For the first problem, you will need to use the following declarations in p1.h.

```
struct point
{
        int x, y;
};

enum direction{NORTH, SOUTH, EAST, WEST};

struct point changeLocation(struct point p, enum direction d);
```

Notice that there is no use of the typedef key word in the above declarations. In your p1.c file, write the function definition for changeLocation. changeLocation will return a new point whose value is the point p, moved in the direction d. Therefore, if d is NORTH, then the y portion is incremented. If d is SOUTH, then the y portion is decremented. If d is EAST or WEST, the x portion is incremented or decremented respectively. So the following separate function calls would return the point whose x and y values are 0.

```
struct point p = {0, -1};
changeLocation(p, NORTH);

struct point p = {0, 1};
changeLocation(p, SOUTH);

struct point p = {1, 0};
changeLocation(p, WEST);

struct point p = {-1, 0};
changeLocation(p, EAST);
```

Put the declarations of these functions into p1.h and put there definitions into p1.c. p1.h will be included from our driver file and tested using our main() function.

2. **(5 Points)** For this problem, you will need to use the following declarations in p2.h.

```
typedef struct
{
        float imag, real;
}ComplexNum;

ComplexNum add(ComplexNum c1, ComplexNum c2);
ComplexNum subtract(ComplexNum c1, ComplexNum c2);
ComplexNum multiply(ComplexNum c1, ComplexNum c2);
ComplexNum divide(ComplexNum c1, ComplexNum c2);
ComplexNum getConjugate(ComplexNum c1);
float getNorm(ComplexNum c1);
```

Note the use of the typedef keyword in the structure declaration. In p2.c, you need to define the functions add, subtract, multiply, divide, getConjugate and getNorm. These functions perform the specified operation on the give complex number or numbers(Note the norm is also referred to as the modulus). If you need a refresher on complex number arithmetic, you may want to check out

http://en.wikipedia.org/wiki/Complex_number#Absolute_value.2C_conjugation_and_distance

or

http://mathworld.wolfram.com/ComplexNumber.html

Put the declarations of these functions into p2.h and put there definitions into p2.c. p2.h will be included from our driver file and tested using our main() function.

3. **(10 Points)** For this problem, you will need to use the following declarations in p3.h.

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

struct node* insertAtBeginning(struct node* first, int valToInsert);
struct node* insertAtEnd(struct node* first, int valToInsert);
int findValue(struct node* first, int valToFind);
struct node* deleteValue(struct node* first, int valToDelete);
void printList(struct node* first);
```

In p3.c, you need to define the functions above. Here we give a description of each function:

struct node* insertAtBeginning(struct node* first, int valToInsert);

This takes a pointer to the first element of a linked list and a value to add to the linked list. A new node is allocated to hold the new value is this node is added to the beginning of the list. It then returns a pointer to the first node in the list. (This should be the newly inserted node.) Note that if first is NULL, it means the list is empty and the given value will be the first and only node in the list when the function returns.

struct node* insertAtEnd(struct node* first, int valToInsert);

This takes a pointer to the first element of a linked list and a value to add to the linked list. A new node is allocated to hold the new value is this node is added to the end of the list. It then returns a pointer to the first node in the list. (This should not change unless the list was originally empty, in which case the first and last node are the same.) Note that if first is NULL, it means the list is empty and the given value will be the first and only node in the list when the function returns.

int findValue(struct node* first, int valToFind);

This function searches through the given linked list, designated by the first pointer. It will return a 1 if it finds a node in the list with the given value to find and a 0 if it does not find a node with the given value in the list.

struct node* deleteValue(struct node* first, int valToDelete);

This function searches through the given linked list, designated by the first pointer. It will remove the first node containing the value valToDelete from the linked list. If it does not find a node containing the given value, then the list is unmodified. It returns the a pointer to the first node in the list after the deletion occurs. (This should be the same as the given pointer, unless the first node contains the value to be deleted.)

void printList(struct node* first);

This function will print the data from each node. It prints it out one value per line.

Note that insertAtBeginning, findValue, and deleteValue have been discussed in lecture slide. Some modifications need to be made to the delete value code, but most of the code can be taken directly from the slides.

Put the declarations of these functions into p3.h and put there definitions into p3.c. p3.h will be included from our driver file and tested using our main() function.

Instructions for submitting Assignment 8

1. First compress your program files into a tar file. For this, change to the directory where you have stored both the files. At the terminal command prompt (putty, Linux console, MAC terminal, etc.) type:

tar cvf assign8.tar p1.h p1.c p2.c p2.h p3.c p3.h

Let's say you have stored your files in assign8 directory on your "rain" machine.

> cd assign8
> ls