CptS 122 Lab #2: Linked Lists

Along with vectors, linked lists are one of the fundamental data structures in computer science. Unlike vectors, which store information in a contiguous block of computer memory, linked lists have the potential to store information in non-adjacent memory blocks. In this lab, you will write a very basic doubly-linked list. To complete this lab, you will need to have created the appropriate .h and .c files that implement the following struct and functions:

```
typedef struct IntList_t
{
    int value;
    struct IntList_t *previous;
    struct IntList_t *next;
} IntList_t;
```

IntList_t* intList_init(int starting_value)

Like our vector's init function, the LL version must dynamically create a new IntList_t and set its starting value as well as setting the previous and next pointers to NULL.

IntList_t * intList_rewind(IntList_t *some_list)

This function returns a pointer to the "beginning" of the linked list. For our purposes, we will consider the beginning of a linked list to be represented by the IntList_t whose previous pointer is equal to NULL.

void intList_add(IntList_t *some_list, int value)

This function will add a new IntList_t to the end of the current linked list. Note that for our purposes, the last element in the list is represented by the individual IntList_t whose next pointer is equal to NULL.

void intList_remove(IntList_t *some_list)

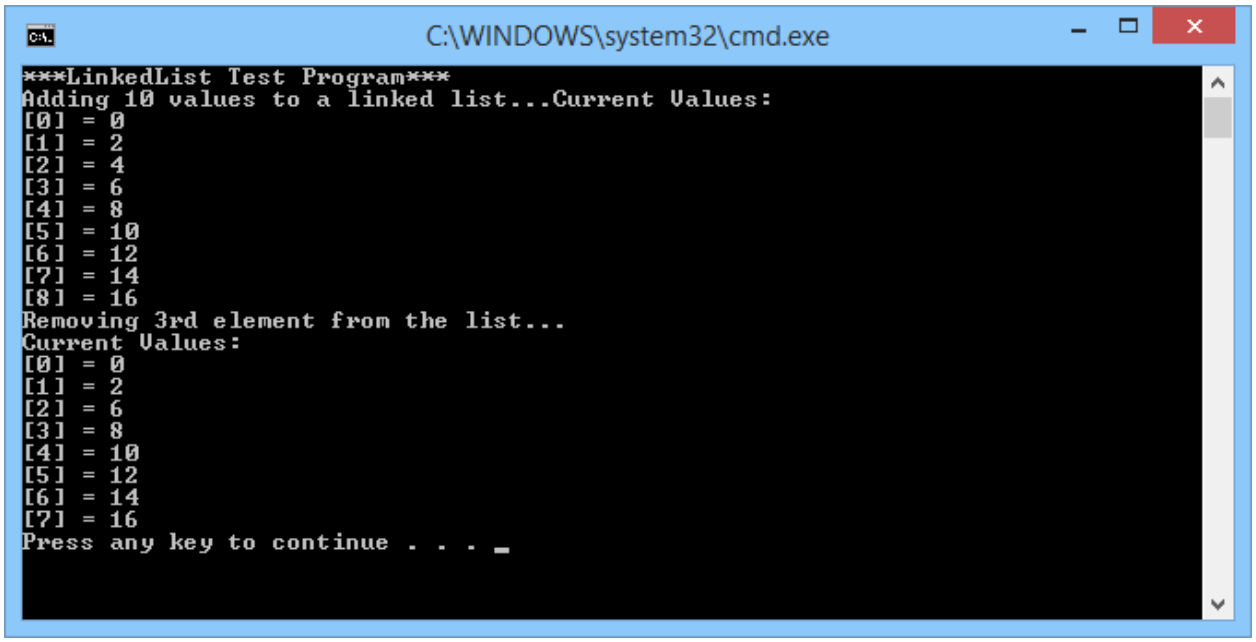
This function removes the supplied IntList_t from the current chain of list items.

void intList_free(IntList_t *some_list)

Freeing up dynamic memory allocated for a linked list is a little more work than freeing up dynamic memory for a vector. As such, this function should call free on every node in the linked list.

Sample Output

To test these methods, you must write a simple test driver in your main function. Here is an example of my test driver:


C:\WINDOWS\system32\cmd.exe
LinkedList Test Program
Adding 10 values to a linked list...Current Values:
[0] = 0
[1] = 2
[2] = 4
[3] = 6
[4] = 8
[5] = 10
[6] = 12
[7] = 14
[8] = 16
Removing 3rd element from the list...
Current Values:
[0] = 0
[1] = 2
[2] = 6
[3] = 8
[4] = 10
[5] = 12
[6] = 14
[7] = 16
Press any key to continue . . .