

COP 2535: Data Structures

Exercise 07, Single Linked List

1 Instructions

Implement a singly linked list. Use pointers and pointer manipulation. This is an open book, open note, and open internet exercise.

2 Template

Implement the code, as indicated. Note that the functions that you should implement both take a pointer as an argument, and return a pointer. I left the display function as a hint for you. Please pay careful attention to the output. You need to code the user prompts, declare the relevant variables, allocate dynamic memory, and implement the function.

You can use the code in the *Mastering C Algorithms* book, and you can use the previous lab code I have given you. If possible, work with another student. The purpose of this exercise is to give you the opportunity to learn how a linked list is implemented from scratch. Don't be afraid to fail, and don't give up. (I speak from experience!)

```

/*****
 * Name:
 * Author:
 * Date:
 * Purpose: Write a program to create a linked list and perform insertions and deletions of all cases.
 * Write functions to sort and finally delete the entire list at once.
 * *****/

#include <iostream>
using namespace std;

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

typedef struct node Node;

//variable declarations
int option;
Node* start = NULL;

//function declarations
int menu();
void dowhat(int);

//these are the functions that you should define below

```

```

Node* create_ll(Node*);
Node* display(Node*);
Node* insert_beg(Node*);
Node* insert_end(Node*);
Node* insert_before(Node*);
Node* insert_after(Node*);

int main()
{
    cout << "Running SingleLinkedList" << endl;
    do
    {
        option = menu();
        //cout << "option is " << option << endl;
        cout << "start is a pointer to" << start << endl;
        dowhat(option);
    } while (option != 13);

    return 0;
}

int menu()
{
    int option;
    cout << "\n\n *****MAIN MENU *****";
    cout << "\n 1: Create a list";
    cout << "\n 2: Display the list";
    cout << "\n 3: Add a node at the beginning";
    cout << "\n 4: Add a node at the end";
    cout << "\n 5: Add a node before a given node";
    cout << "\n 6: Add a node after a given node";
    cout << "\n 13: EXIT";
    cout << "\n Enter your option: ";
    cin >> option;
    //cout << "You chose " << option << endl;
    return option;
}

void dowhat(int option)
{
    switch (option)
    {
        case 1: start = create_ll(start);
            cout << "\n LINKED LIST CREATED";
            break;
        case 2: start = display(start);
            break;
        case 3: start = insert_beg(start);
            break;
        case 4: start = insert_end(start);
            break;
        case 5: start = insert_before(start);
            break;
        case 6: start = insert_after(start);
    }
}

```

```

        break;
    }
}

Node* create_ll(Node* start)
{
    cout << "Calling create_ll()\n";
    //implement your code here

    return start;
}

Node* display(Node* start)
{
    cout << "Calling display()\n";
    Node* ptr;
    ptr = start;
    cout << "\nLinked List ----- \n";
    cout << " - start->%p" << start << endl;
    while (ptr != NULL)
    {
        cout << " - %p<-%d->%p: " << ptr << " | " << ptr->data << " | " << ptr->next << endl;
        ptr = ptr->next;
    }
    cout << "\n----- \n";
    return start;
}

Node* insert_beg(Node* start)
{
    cout << "Calling insert_beg()\n";
    //implement your code here
    return start;
}

Node* insert_end(Node* start)
{
    cout << "Calling insert_end()\n";
    //implement your code here
    return start;
}

Node* insert_before(Node* start)
{
    cout << "Calling insert_before()\n";
    //implement your code here
    return start;
}

Node* insert_after(Node* start)
{
    cout << "Calling insert_after()\n";
    //implement your code here
    return start;
}

```

```
}
```

3 Output

In my run, I took the following steps:

- I created a list with 2, 4, 6, and 8, and displayed the list
- I added 66 to the beginning of the list and displayed the list
- I added 77 to the end of the list and displayed the list
- I added 88 before the value 6 and displayed the list
- I added 99 after the value 6 and displayed the list

Note that the `print` statements are designed to help you understand that is going on in the computer.

Your deliverable consists of your source code and your output file. Your deliverable is a text file that will look similar to this. You should be able to select your output screen, copy it (with Ctl-C), and insert it into the text entry box (Ctl-V).

Running SingleLinkedList

```
*****MAIN MENU *****
1: Create a list
2: Display the list
3: Add a node at the beginning
4: Add a node at the end
5: Add a node before a given node
6: Add a node after a given node
13: EXIT
Enter your option: 1
start is a pointer to000000000000000000
Calling create_ll()
Enter an integer, -1 to end: 2
Enter an integer, -1 to end: 4
Enter an integer, -1 to end: 6
Enter an integer, -1 to end: 8
Enter an integer, -1 to end: -1

LINKED LIST CREATED

*****MAIN MENU *****
1: Create a list
2: Display the list
3: Add a node at the beginning
4: Add a node at the end
5: Add a node before a given node
6: Add a node after a given node
13: EXIT
Enter your option: 2
start is a pointer to000001FDAB2ED100
Calling display()

Linked List -----
```

```

- start->%p000001FDAB2ED100
- %p<-%d->%p: 000001FDAB2ED100 | 2 | 000001FDAB2ED150
- %p<-%d->%p: 000001FDAB2ED150 | 4 | 000001FDAB2ED1A0
- %p<-%d->%p: 000001FDAB2ED1A0 | 6 | 000001FDAB2ED1F0
- %p<-%d->%p: 000001FDAB2ED1F0 | 8 | 0000000000000000

```

*****MAIN MENU *****

```

1: Create a list
2: Display the list
3: Add a node at the beginning
4: Add a node at the end
5: Add a node before a given node
6: Add a node after a given node
13: EXIT

```

Enter your option: 3

start is a pointer to 000001FDAB2ED100

Calling insert_beg()

Enter an integer: 66

in insert_beg() --- new_node address: %p, new_node->data: %d, new_node->next: %p
000001FDAB2ECAC0 | 66 | 000001FDAB2ED100

*****MAIN MENU *****

```

1: Create a list
2: Display the list
3: Add a node at the beginning
4: Add a node at the end
5: Add a node before a given node
6: Add a node after a given node
13: EXIT

```

Enter your option: 2

start is a pointer to 000001FDAB2ECAC0

Calling display()

Linked List -----

```

- start->%p000001FDAB2ECAC0
- %p<-%d->%p: 000001FDAB2ECAC0 | 66 | 000001FDAB2ED100
- %p<-%d->%p: 000001FDAB2ED100 | 2 | 000001FDAB2ED150
- %p<-%d->%p: 000001FDAB2ED150 | 4 | 000001FDAB2ED1A0
- %p<-%d->%p: 000001FDAB2ED1A0 | 6 | 000001FDAB2ED1F0
- %p<-%d->%p: 000001FDAB2ED1F0 | 8 | 0000000000000000

```

*****MAIN MENU *****

```

1: Create a list
2: Display the list
3: Add a node at the beginning
4: Add a node at the end

```

```

5: Add a node before a given node
6: Add a node after a given node
13: EXIT
Enter your option: 4
start is a pointer to 000001FDAB2ECAC0
Calling insert_end()

```

```

Enter an integer: 77
in insert_end() --- new_node address: %p, new_node->data: %d, new_node->next: %p | 000001FDAB2ED240 | 77

```

```

*****MAIN MENU *****
1: Create a list
2: Display the list
3: Add a node at the beginning
4: Add a node at the end
5: Add a node before a given node
6: Add a node after a given node
13: EXIT
Enter your option: 2
start is a pointer to 000001FDAB2ECAC0
Calling display()

```

```

Linked List -----
- start->%p000001FDAB2ECAC0
- %p<-%d->%p: 000001FDAB2ECAC0 | 66 | 000001FDAB2ED100
- %p<-%d->%p: 000001FDAB2ED100 | 2 | 000001FDAB2ED150
- %p<-%d->%p: 000001FDAB2ED150 | 4 | 000001FDAB2ED1A0
- %p<-%d->%p: 000001FDAB2ED1A0 | 6 | 000001FDAB2ED1F0
- %p<-%d->%p: 000001FDAB2ED1F0 | 8 | 000001FDAB2ED240
- %p<-%d->%p: 000001FDAB2ED240 | 77 | 0000000000000000

```

```

*****MAIN MENU *****
1: Create a list
2: Display the list
3: Add a node at the beginning
4: Add a node at the end
5: Add a node before a given node
6: Add a node after a given node
13: EXIT
Enter your option: 5
start is a pointer to 000001FDAB2ECAC0
Calling insert_before()

```

```

Enter an integer: 88

Enter the value before which the data has to be inserted: 6
in insert_before() --- new_node address: %p, new_node->data: %d, new_node->next: %p
000001FDAB2F8BC0 | 88 | 000001FDAB2ED1A0

```

```

*****MAIN MENU *****
1: Create a list
2: Display the list
3: Add a node at the beginning
4: Add a node at the end
5: Add a node before a given node
6: Add a node after a given node
13: EXIT
Enter your option: 2
start is a pointer to 000001FDAB2ECAC0
Calling display()

```

```

Linked List -----
- start->%p000001FDAB2ECAC0
- %p<-%d->%p: 000001FDAB2ECAC0 | 66 | 000001FDAB2ED100
- %p<-%d->%p: 000001FDAB2ED100 | 2 | 000001FDAB2ED150
- %p<-%d->%p: 000001FDAB2ED150 | 4 | 000001FDAB2F8BC0
- %p<-%d->%p: 000001FDAB2F8BC0 | 88 | 000001FDAB2ED1A0
- %p<-%d->%p: 000001FDAB2ED1A0 | 6 | 000001FDAB2ED1F0
- %p<-%d->%p: 000001FDAB2ED1F0 | 8 | 000001FDAB2ED240
- %p<-%d->%p: 000001FDAB2ED240 | 77 | 0000000000000000
-----

```

```

*****MAIN MENU *****
1: Create a list
2: Display the list
3: Add a node at the beginning
4: Add a node at the end
5: Add a node before a given node
6: Add a node after a given node
13: EXIT
Enter your option: 6
start is a pointer to 000001FDAB2ECAC0
Calling insert_after()

```

Enter an integer: 99

```

Enter the value after which the data has to be inserted: 6
in insert_after() --- new_node address: %p, new_node->data: %d, new_node->next: %p: 000001FDAB2F8800 | 9

```

```

*****MAIN MENU *****
1: Create a list
2: Display the list
3: Add a node at the beginning
4: Add a node at the end
5: Add a node before a given node
6: Add a node after a given node
13: EXIT
Enter your option: 2
start is a pointer to 000001FDAB2ECAC0
Calling display()

```

Linked List -----

```
- start->%p000001FDAB2ECAC0
- %p<-%d->%p: 000001FDAB2ECAC0 | 66 | 000001FDAB2ED100
- %p<-%d->%p: 000001FDAB2ED100 | 2 | 000001FDAB2ED150
- %p<-%d->%p: 000001FDAB2ED150 | 4 | 000001FDAB2F8BC0
- %p<-%d->%p: 000001FDAB2F8BC0 | 88 | 000001FDAB2ED1A0
- %p<-%d->%p: 000001FDAB2ED1A0 | 6 | 000001FDAB2F8800
- %p<-%d->%p: 000001FDAB2F8800 | 99 | 000001FDAB2ED1F0
- %p<-%d->%p: 000001FDAB2ED1F0 | 8 | 000001FDAB2ED240
- %p<-%d->%p: 000001FDAB2ED240 | 77 | 0000000000000000
```

*****MAIN MENU *****

```
1: Create a list
2: Display the list
3: Add a node at the beginning
4: Add a node at the end
5: Add a node before a given node
6: Add a node after a given node
13: EXIT
```

Enter your option: 13

start is a pointer to 000001FDAB2ECAC0

C:\Users\ccc31\CPlusPlus\SingleLinkedList02\x64\Debug\SingleLinkedList02.exe (process 13008) exited with
Press any key to close this window . . .