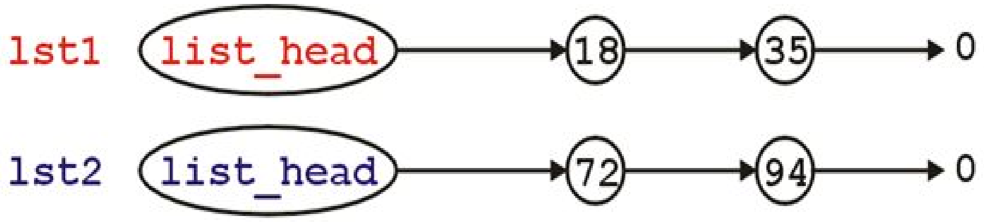
Name:
Login:
Signature:

ECE 368 Spring 2016.
Homework 3

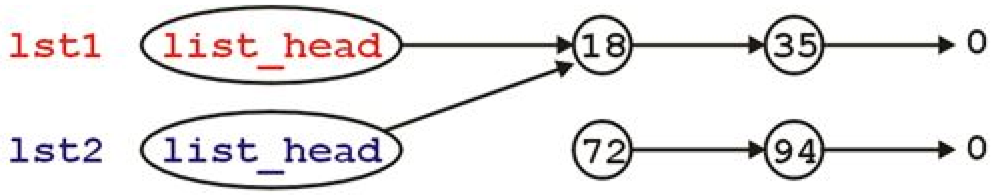
1) Linked Lists Assignment in C++ (15pts).

Consider the two linked lists explained in lecture 10, slide 4: lst1 and lst2.

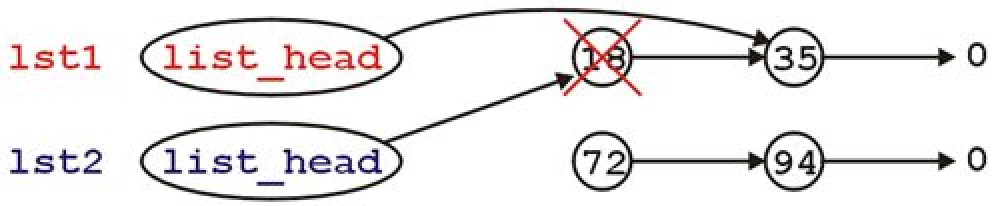


If we try to assign lst2 to lst1, we will run into problems because lst2 will point only to the head of lst1 (because of the way the linked list class is defined.)

```
lst2 = lst1;
```



We have 2 problems, First: if we call lst1.pop_front(); lst2 will point to null. Second, the elements 72 and 94 are wasted (shown in slide 5 of lecture 10).



(Continues in the next page)

1) **Given a class defined in an object oriented programming language like C++ or Python, explain what is “Operator Overloading”.**

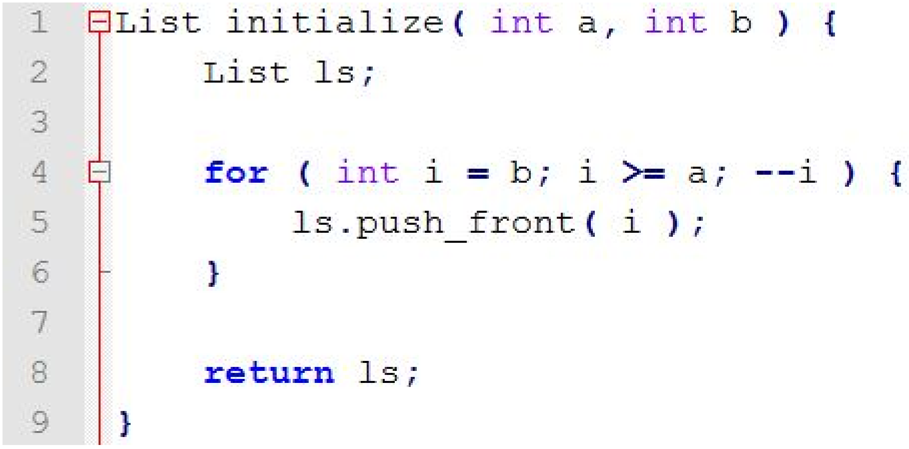
2) **Write a piece of C++ code to fix these problems when performing the following operation: lst2 = lst1 (Hint: Use Operator Overloading).**

2) Fixing a linked list constructor (15pts).

Consider Slide 8 of Lecture 10. We construct a linked list with the command:

List vec = initialize(3, 6);

Where initialize is defined as follows.



```
List initialize( int a, int b ) {
    List ls;

    for ( int i = b; i >= a; --i ) {
        ls.push_front( i );
    }

    return ls;
}
```

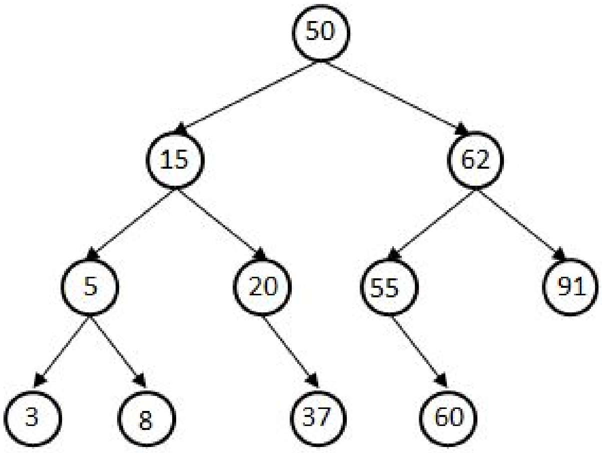
However, the function **initialize(int a, int b)** will destruct the local variable ***ls*** after execution. Therefore, both ***ls*** and ***vec*** will point at null. Your task is to write the **initialize_v2(int a, int b)** constructor so that ***vec*** will point to 3 and ***ls*** will point to null after the function **initialize_v2** is called. (Note: It must be written in C++)

3) Reversing a Linked List (15pts).

Write a function (in C++ or C) to reverse a singly linked list. The function takes a pointer to a list head as input parameter and returns a pointer to the reversed list new head:

4) Binary Search Tree (15pts).

The following diagram shows a binary search tree. Perform the following operations: insert 24, insert 19, and delete 50. Use the inorder successor to perform node deletion. (Draw the new BST after every operation)



5) Programming Assignment (40pts).

Implement the following traversal algorithms: Inorder, Preorder and Postorder. The files “hw3.c” and “test.txt” will be provided through the assignment on blackboard. Submit only one c file with the name: “hw3_sol.c”.

Given the sample text file “test.txt,” your code should have the following output:

```
Inorder Traversal: 1 2 3 4 5 6 7 8 9 10 11 12
Preorder Traversal: 4 2 1 3 5 10 6 8 7 9 12 11
Postorder Traversal: 1 3 2 7 9 8 6 11 12 10 5 4
```