Instructions:

- You must submit your work within 2 hours(late submissions are not allowed)
- You should name your cpp file in this format rollNum.cpp i.e. 1348.cpp for roll num 20L-1348
- Cheating is prohibited

NOTE:

For both the Questions you can use following Node struct and the creation method

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


Node * createNode (int data)
{
  Node *n = new Node ();
  n->data = data;
  n->next = NULL;
  return n;
}

//TO TEST YOUR CODE PRINT FUNCTION(for your testing purpose only)

void Print (Node * head)
{
  while (head != NULL)
    {
      cout << head->data << " ";
      head = head->next;
    }
}
```

Announcement:

You don't need to implement a linked list class, your task is only to implement the functions for both the questions. You may make as many helper functions as you want to use in the following functions.

Question 1

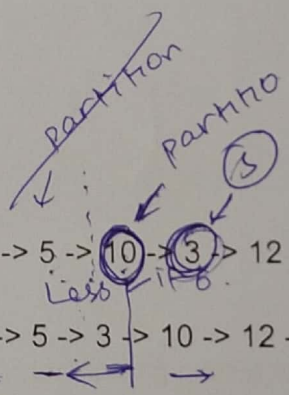
There's a requirement for a warehouse that whatever they receive should be converted to an order based on some integer number(X) that the warehouse manager will decide. So, considering this your task is to implement a function that takes LinkedList's head containing the integer data item for each node and partition the linked list in such a way that all nodes less than X comes first , then all nodes with equal value to X, and finally all nodes with greater than X. **Make sure the relative order of nodes in all partitions should be preserved.**

1 4 2 5 10 3 12 15

Function Declaration:

```
Node* getPartitionedList(Node* head, int x){
//your code goes here
}
```

Time Complexity: O(N)



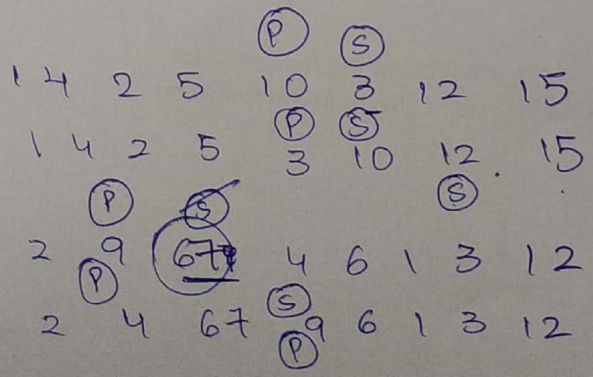
Example 1:

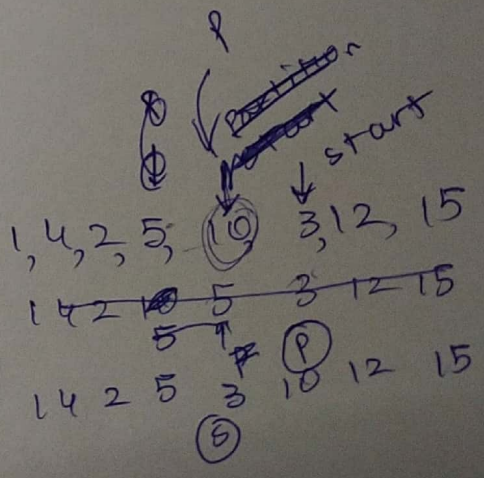
Linked List: 1 -> 4 -> 2 -> 5 -> 10 -> 3 -> 12 -> 15
X: 6
Resultant : 1 -> 4 -> 2 -> 5 -> 3 -> 10 -> 12 -> 15

In this example note that order of nodes is preserved i.e. Nodes are appearing in order. All nodes smaller than 6 are in the order of the original linked list i.e. 1 -> 4 -> 2 -> 5 -> 3 and greater are 10->12->15.

Example 2:

Linked List: 2 -> 9 -> 67 -> 4 -> 6 -> 1 -> 3 -> 12
X : 4
Linked List: 2 -> 1 -> 3 -> 4 -> 9 -> 67 -> 6 -> 12





Question 2:

At an intelligence firm there are long codes in a form of singly linked list (every node containing an integer data) associated with each employee and in case of any urgent or unforeseen situation an employee has to type the code in a short form by **removing the consecutive duplicate integers with the total consecutive integer count** so that the department knows that the employee is in an unforeseen situation. So your job is to assist the intelligence department by writing a function **Node* simplifyCode(Node* head)** that takes a singly linked list head as an argument and returns the simplified code linked list head.

Function declaration:

```
Node* simplifyCode(Node* head){
//your code goes here
}
```

Time Complexity: O(N)

Example1:

Input: s = 1->1->0->2->2->2->2

Output List: 1->2->0->2->4

Example2:

Input:1->1->1->1->1->2->1

Output: 1->5->2->1