Lab 8

1 Objectives

The purpose of this lab is to reinforce dynamic container class concepts and linked lists in C++. The labs consists of following problems:

2 Requirements

2.1 Print a range

Write a bag member function with two parameters. The two parameters are Items x and y. The function should write to the console all Items in the bag that are between the first occurrence of x and the first occurrence of y. You may assume that items can be compared for equality using ==.

Use the following header for the function:

void print_value_range(const Item& x, const Item& y);

print_value_range can be interpreted in a number of ways, but use the following points. This should make the implementation a little easier.

1. Print the Items from x to y including the start but not including the end.
2. If there is no element in the bag that has value x, print nothing
3. Otherwise, if there is no element in the bag, after x, that has the value y, then print from x to the end of the list
4. Print the values on one line separated by space. Put an end of line after the values are all printed.

Here are some examples:

Bag [1,2,3,4,5,6,7]

x = 2

y = 5

prints 2 3 4

Bag [1,2,3,4,5,6,7]

x = 2

y = 78

prints 2 3 4 5 6 7

Bag [1,2,3,4,5,6,7]

x = 2

y = 1

prints 2 3 4 5 6 7

Bag [1,2,3,4,5,6,7]

x = 8

y = 5

prints (nothing)

2.2 Remove repetitions

Write a member function that deletes all repetitions from the bag. In your implementation, assume that items can be compared for equality using ==. Use the following header for the function:

void remove_repetitions()

Here is a brief outline of an algorithm:

1. A node pointer p steps through the bag
2. For each Item, define a new pointer q equal to p
3. While the q is not the last Item in the bag
    - If the next Item has data equal to the data in p, remove the next Item
    - Otherwise move q to the next Item in the bag

2.3 Write test program to test the above two member functions