**1. General**

**Problem description**

Developed and tested a class B+Tree to store pairs of the form (key, value). Implement a memory resident B+tree (i.e., the entire tree resides in main memory) that was able to store multiple pairs that have the same key (i.e., duplicates). The leaves were linked into a doubly linked list to support efficient range retrieval. The supported operations are:

1. Initialize(m): create a new order m B+Tree

2. Insert (key, value)

3. Search (key) : returns all values associated with the key

4. Search (key1,key2): returns (all key value pairs) such that key1 <= key <= key2.

**Programming Environment**

Java

**2. Input and Output Requirements**

**Input Format**

The first line in the input file contains the order (m) of the B+Tree. Each of the remaining lines specifies a B+tree operation. The following is an example of an input file.

12

Insert(3.55,Value1)

Insert(4.01,Value10)

Insert(39.56,Value2)

Insert(-3.95,Value23)

Insert(-3.91,Value54609)

Insert(3.55,Value67)

Insert(0.02,Value98)

Search(3.55)

Search(-3.91,30.96)

Insert(3.26,Value56089)

Insert(121.56,Value1234)

Insert(-109.23,Value43234)

Search(3.71)

You can use float/double as the type of the key and string as the type of the value.

**Output Format**

For an Insert query you should not produce any output.

For a Search query you should output the results on a single line using commas to separate values. The output for each search query should be on a new line. All output should go to a file named “output\_file.txt”.

For a range search query, the output should be sorted according to the key (smallest to largest). Output should contain (key,value) pairs separated by commas.

If there are multiple values for some key, these may be output in any order. If a search query does not return anything you should output “Null”.

The following is the output file for the above input file.

Value 67, Value1

(-3.91,Value54609), (0.02,Value98), (3.55,Value1), (3.55,Value67), (4.01, Value10)

Null