National University of Computer and Emerging Sciences, Lahore Campus

	Course:	Data Structures	Course Code:	CS2001
	Program:	BS (Data Science)	Semester:	Fall 2022
	Deadline	6-Nov-22	Total Marks:	20
	Section:	A, B	Page(s):	2

Important Instructions:
1. Submit your source files in a zipped file named as your roll number, i.e., 20L-1111.zip.
2. You are not allowed to copy solutions from other students. We will check your code for plagiarism using plagiarism checkers. If any sort of cheating is found, negative marks will be given to all students involved.
3. Late submission of your solution is not allowed.

Introduction:

Search engines have become an integral part of our lives living in this technological day and age. In this assignment you will be implementing your very own search engine. Your program should be able to read a database from a file and allow it to be searched. It should be based on a command line interface. The code should be clean and properly commented.

Input Format:

A sample file is provided to you (data.txt). In this sample file, every pair of lines is a data item. The first of these lines contains an ID and a URL. The second line contains a set of keywords (space separated). The ID will not be required by the search engine, so you can simply ignore it. There will be a blank line between pairs of lines.

Search Engine:

Implement the search engine using a BST (binary search tree). The basic idea here is to have each node store a keyword, along with associated list of URLs. It would be helpful to implement your BST (binary search tree) using a class, with left and right child pointers, similar to what was done in the lectures. All you have to do here is implement the **insert function, search function, delete function and the destructor** that were explained. Your tree node could be something like:

```
Class BSTNode
{
string keyword; //key of the tree node
list<string> URLs;
BSTNode *lChild;
BSTNode *rChild;
}
```

To store the URLs, you can either use list or vector class available in C++. The list class is actually a doubly linked list, and the vector class is actually a dynamic array. You can also use your own implemented linked list (singly or doubly) or dynamic array to store the URLs if you want.

- The signature of search function of your tree could look like this:
  list<string> * search(string keyword);

- The signature of insert function of your tree could look like this:
  void insert(string keyword, string url); (if the keyword does not exist, then create a new node with the given keyword as key, and add the url in the list. If the keyword already exists, then simply append the URL in the existing node's list. Also add a check for duplicate URL.)

- The signature of delete function could look like this:
  void delete(string keyword); (Delete the node containing the given keyword)

Main Function:
At the start, your main function should ask for a filename. It will then read the data from that file, load it into the binary search tree that you made in the previous step, and output the time taken to load the data from the file into the BST. It then should ask for a keyword and output the list of URLs that match it. It should also output the time taken for the search. The program will continue to ask for the word to search until the user enters the minus key.

A sample run of your program is given below. Bold lines indicate program output, lines starting with a '>' indicate user entered input.

Please enter a filename:
> data.txt
File loaded into the BST successfully. 53 URLs loaded from the file in 0 seconds and 548 microseconds.

Please enter a word to search, or enter - to exit:
> life

3 result(s) found in 0 seconds and 431 microseconds.
1. http://www.gutenberg.net/dickens/otwist/4.html
2. http://www.gutenberg.net/dickens/otwist/32.html
3. http://www.gutenberg.net/dickens/otwist/40.html

Please enter a word to search, or enter - to exit:
> -

(This assignment has been adapted from CS202 at LUMS)