Indian Institute of Technology Roorkee

Department of Computer Science and Engineering

CSN-261: Data Structures Laboratory (Autumn 2019-2020)

Lab Assignment-4 (L4) **Date: August 21, 2019** **Duration: 2 Weeks**

General Instructions:

1. Every Lab Assignment will be performed by the students individually. No group formation is required and the evaluations will be done every week for the students individually.
2. The student should use Deoxygen tool (http://www.doxygen.nl) for getting automatic documentation of the codes written by him/her.
3. For version control of the written codes, the students are being instructed to learn and use any open-source CSV tool (https://www.nongnu.org/cvs) or GitHub (https://github.com).

Submission and Evaluation Instructions:

1. Submit your zipped folder (**<filename>.zip** or **<filename>.tar.gz**) through your account in Moodle through the submission link for this Lab Assignment in Moodle course site: https://moodle.iitr.ac.in/course/view.php?id=46.
2. **Hard deadline for Final submission in Moodle**: **September 04, 2019 (1:00 pm Indian Time).** For any submission after Final Deadline, 20% marks will be deducted (irrespective of it is delayed by a few seconds or a few days). The key to success is starting early. You can always take a break, if you finish early.
3. The submitted zipped folder (**<filename>.zip** or **<filename>.tar.gz**) must contain the following:
   (a) The source code files in a folder
   (b) A report file (**<filename>.DOC** or **<filename>.PDF**) should contain the details like:
      i. Title page with details of the student
      ii. Problem statements
      iii. Algorithms and data structures used in the implementation
      iv. Snapshots of running the codes for each problem
4. The submission by each student will be checked with others' submission to identify any copy case (using such detection software). If we detect that the code submitted by a student is a copy (partially or fully) of other's code, then the total marks obtained by one student will be divided by the total number of students sharing the same code.

Instructions for L4:

1. Objective of this Lab Assignment L4 is to make the students familiar with different data structures while coding the programs in the C++ language to solve some real-life problems, using STL (Standard Template Library), Qt library and creating an installer (for Windows OS) of your GUI based programs.
2. The students are expected to have a basic knowledge of data structures and the C++ programming language.
3. The student will have to demonstrate and explain the coding done for this Lab Assignment L4 in the next laboratory class to be held on **September 04, 2019** for evaluation.

Problem Statement 1:

Create a dictionary using Trie data structure (**without using STL**) having words and their meanings. You need to read the words and their respective meanings from a CSV file (uploaded in Piazza, named as *TrieInput.csv*), where 1st column is for words and 2nd column shows its meaning.

Given a word you have to print its meaning. If no such word is found in the dictionary, then print "Invalid word". Create a GUI using Qt library to accept a word in a text box and display the meaning in an another box, as shown in the Figure 1.

Also, create an installer of your program for Windows OS. You can use the software like InstallSimple or InstallShield or WIX or NSIS to do so.

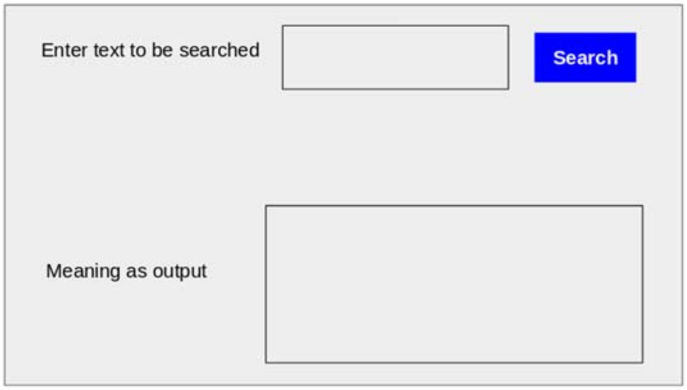

Figure 1: GUI for dictionary using Trie data structure.

Problem Statement 2:

Implement N Queens problem to show all the possible combinations in N x N binary matrix and to display the total number of such combinations at the end, where 1 represents the position of N queens in the N x N matrix and remaining cells are represented by 0.

A sample output for N=4 is shown below.

Input:

N: 4

Output:

Combination 1:

0	1	0	0
0	0	0	1
1	0	0	0
0	0	1	0

Combination 2:

0	0	1	0
1	0	0	0
0	0	0	1
0	1	0	0

Total number of combinations: 2

Problem Statement 3:

Given an integer array having N number of elements, write a C++ program using hash map (**using STL**) to find the length of the largest subarray from the given input array, where the summation of the elements of the subarray is equal to *n*. In the output, if any solution exists then print the starting and ending index (with respect to given input array) of the largest subarray and also print its length. Otherwise, print “Not Found”, as described in the following output.

Input:

N = 8
15 0 2 -3 1 5 3 -2
n = 5

Output:

Length of longest subarray is 5
Index from 1 to 5.