National Institute of Technology, Calicut
Department of Computer Science and Engineering
CS2094 – Data Structures Lab
Assignment-1

Submission deadline (on or before):
17th January 2016, 10:00:00 PM (for both Main and Advanced batches)

Policies for Submission and Evaluation

You must submit your assignment in the moodle (Eduserver) course page, on or before the submission deadline. Also, ensure that your programs in the assignment must compile and execute without errors in Athena server. During evaluation your uploaded programs will be checked in Athena server only. Failure to execute programs in the assignment without compilation errors may lead to zero marks for that program.

Your submission will also be tested for plagiarism, by automated tools. In case your code fails to pass the test, you will be straightaway awarded zero marks for this assignment and considered by the examiner for awarding F grade in the course. Detection of ANY malpractice regarding the lab course will also lead to awarding an F grade.

Naming Conventions for Submission

Submit a single ZIP (.zip) file (do not submit in any other archived formats like .rar or .tar.gz). The name of this file must be ASSG<NUMBER>_<ROLLNO>_<FIRST-NAME>.zip (For example: ASSG1_BxxyyyyCS_LAXMAN.zip). DO NOT add any other files (like temporary files, input files, etc.) except your source code, into the zip archive.

The source codes must be named as ASSG<NUMBER>_<ROLLNO>_<FIRST-NAME>_<PROGRAM-NUMBER>.<extension> (For example: ASSG1_BxxyyyyCS_LAXMAN_1.c). If there is a part *a* and a part *b* for a particular question, then name the source files for each part separately as in ASSG1_BxxyyyyCS_LAXMAN_1b.c.

If you do not conform to the above naming conventions, your submission might not be recognized by some automated tools, and hence will lead to a score of 0 for the submission. So, make sure that you follow the naming conventions.

Standard of Conduct

Violations of academic integrity will be severely penalized.

Each student is expected to adhere to high standards of ethical conduct, especially those related to cheating and plagiarism. Any submitted work MUST BE an individual effort. Any academic dishonesty will result in zero marks in the corresponding exam or evaluation and will be reported to the department council for record keeping and for permission to assign F grade in the course. The department policy on academic integrity can be found at:
http://cse.nitc.ac.in/sites/default/files/Academic-Integrity.pdf.

Assignment Questions

General Instructions for all the questions:

- Invalid input should be detected and suitable error messages should be generated.
- Sample inputs are just indicative.

1. Write a program that reads two strings, say s1 and s2, consisting of lower case letters from the English alphabet [a-z] as input and prints whether they are anagrams or not. Given two strings s1 and s2, we say s1 is an anagram of s2, if s2 can be obtained by rearranging the letters of s1 and s2 need not be a meaningful string in the English dictionary.

   Input: 2 strings, **str1** and **str2**, of length at most 100000.
   Output: YES, if str1 is an anagram of str1. NO, otherwise.

   Example:
   Input:

   ```
   Enter str1: lesson
   Enter str2: snoesl
   ```

   Output:

   ```
   YES
   ```

   Input:

   ```
   Enter str1: tonsil
   Enter str2: silent
   ```

   Output:

   ```
   NO
   ```

2. Write a recursive program that reads a positive integer (in decimal form) as input, and prints the number of 1s in its binary representation.

   Input: A positive decimal integer, **num**, in the range 0-2^{31}
   Output: Number of 1s in the binary representation of **num**.

   Example:
   Input:

   ```
   Enter a positive decimal number: 34
   ```

   Output:

   ```
   The binary representation of 34 contains 2 1s.
   ```

3. Create a structure that stores the following details:
   STUDENT-ID
   STUDENT-NAME
   MARK

MARK can either be positive integers in the range 0-100, 'A' for absent or 'I' for insufficient attendance. Create a file by reading the name of the file from the command line. Write a menu driven program that uses this structure and allows the user to perform the following operations:

   a. Add student record into the file.
   b. Display the names of students who obtained less than 30 marks.
   c. Display the names of students who have insufficient attendance.
   d. Print the contents of the file.

You must read the records of the file into the structure and only operate on this structure for the operations *b and c*.

Example:
Input:

```
    1. Add student details
    2. Display the names of students who obtained less than 30 marks
    3. Display the names of students who have insufficient attendance
    4. Display all the records in the file
    5. Exit
       Enter your choice: 1
       Enter name, id and mark:
       Alice
       101AA
       70

    1. Add student details
    2. Display the names of students who obtained less than 30 marks
    3. Display the names of students who have insufficient attendance
    4. Display all the records in the file
    5. Exit
       Enter your choice: 1
       Enter name, id and mark:
       Bob
       102AB
       27

    1. Add student details
    2. Display the names of students who obtained less than 30 marks
    3. Display the names of students who have insufficient attendance
    4. Display all the records in the file
    5. Exit
       Enter your choice: 1
       Enter name, id and mark:
       Dylan
       103BB
       I

    1. Add student details
    2. Display the names of students who obtained less than 30 marks
    3. Display the names of students who have insufficient attendance
    4. Display all the records in the file
    5. Exit
       Enter your choice: 1
       Enter name, id and mark:
       John
       104AB
       23

    1. Add student details
    2. Display the names of students who obtained less than 30 marks
    3. Display the names of students who have insufficient attendance
    4. Display all the records in the file
    5. Exit
       Enter your choice: 2
       Students who scored less than 30:
       Bob
       John
```

```
1. Add student details
2. Display the names of students who obtained less than 30 marks
3. Display the names of students who have insufficient attendance
4. Display all the records in the file
5. Exit
   Enter your choice: 3
   Students with insufficient attendance:
   Dylan

1. Add student details
2. Display the names of students who obtained less than 30 marks
3. Display the names of students who have insufficient attendance
4. Display all the records in the file
5. Exit
   Enter your choice: 4
   +----------+------------+----+
   |Student_ID|Student_Name|Mark|
   +----------+------------+----+
   |101AA     |Alice       |70  |
   |102AB     |Bob         |27  |
   |103BB     |Dylan       |I   |
   |104AB     |John        |23  |
   +----------+------------+----+
```

Note: Output should be as per the format given in sample input and output.

4. Write a program that reads an *n×n* matrix *M* as input and prints the resultant matrix after applying an averaging filter on *M*. The averaging filter is applied on M as follows: For each element $M_{i,j}$, the filter computes the average of itself and all its neighbours, rounded to the nearest integer.

   Note: Neighbours of an element $M_{i,j}$ are $M_{i-1,j-1}$, $M_{i,j-1}$, $M_{i+1,j-1}$, $M_{i-1,j}$, $M_{i+1,j}$, $M_{i-1,j+1}$, $M_{i,j+1}$ and $M_{i+1,j+1}$. All these neighbours may not exist for every element of *M*. For instance, the corner elements of *M* have only 3 neighbours.

   Input: An *n×n* matrix **M**, with 0<n<=100, containing integers in the range 0-2^{31}.
   Output: Matrix M with an averaging filter applied on it.

   Example:
   Input:

   ```
   1 8 2
   2 9 6
   1 3 0
   ```

   Output:

   ```
   5 5 6
   4 4 5
   4 4 5
   ```

5. Write a menu driven calculator program that reads two real numbers as input and performs mathematical operations, namely, addition, subtraction, multiplication and division, using function pointers. The program should use the menu shown in the example given below.

   Input: Two real numbers, and a choice.
   Output: Result of the chosen mathematical operation on the input numbers.

Example:
Input:

```
Enter the numbers: 14 8
Menu:
1.   Add
2.   Subtract
3.   Multiply
4.   Divide
5.   Exit
Enter the choice: 4
1.75

Menu:
1.   Add
2.   Subtract
3.   Multiply
4.   Divide
5.   Exit
Enter the choice: 2
6

Menu:
1.   Add
2.   Subtract
3.   Multiply
4.   Divide
5.   Exit
Enter the choice: 5
```

****