National Institute of Technology Calicut
Department of Computer Science and Engineering
B. Tech. (CSE) – Third Semester
CS2092D: Programming Laboratory
Assignment – 2

Submission deadline (on or before):
16th September 2020, 10:00 PM

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 compile and execute without errors in linux platform. During evaluation, failure to execute programs, in the assignment without compilation errors may lead to zero marks for that program. 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, .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 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 marks 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.

General Instructions
Programs should be written in C language and compiled using C compiler in Linux platform. Invalid input should be detected and suitable error messages should be generated. Sample inputs are just indicative.

<u>Submit the solutions to questions 1, 2, and 3 through the submission link in Eduserver</u>. The remaining questions are given for practice.

<u>QUESTIONS</u>

1. Write a program to read two integers and find the GCD of the two integers (the largest integer that divides each of the integers) using recursion.

   Input format:

   - Input contains two integers separated by a single space.

   Output format:

   - The output is the integer which is the GCD of two integers.
   - As GCD(0,0) is undefined, at most one of the input integers can be zero. If both integers are 0, print -1.

   Sample Input1: 366 60

   Sample Output 1: 6

   Sample Input 2: -10 15

   Sample Output 2: 5

   Sample Input 3: 0 15

   Sample Output 3: 15

   Sample Input 4: 0 0

   Sample Output 4: -1

2. Write a program to print the factorial of a given integer ***n***, using recursion.

   Input format: Input contains an integer $n \in$ **[0, 20]** for which the factorial is to be found.

   Output format: The output is the factorial of the given number ***n***.

   Sample Input: 5

   Sample Output: 120

3. Write a program that reads an array **A** of **n** integers in the ascending order and check whether a given integer ***x*** is present in the array using **recursive binary search**. Assume that the array index starts from 0. If the element ***x*** is present in the array, print the position of the element in the array. If the element **x** presents more than once in the array then print the first occurrence of the element in the array. Otherwise, print -1.

Input format:

- The first line of the input contains an integer $n \in [0, 10^9]$, the size of an array ***A***.
- The second line contains ***n***, space-separated integers in the range **[−1000, 1000].** These elements form the array ***A***.
- The third line contains an integer $x \in [-1000, 1000]$ to be searched in the array.

Output format:

- If ***x*** is present in ***A***, print the index of ***x*** in ***A***.
- If ***x*** is not present in ***A***, print −1.

Sample Input 1:

7

12 35 50 59 60 73 90

73

Sample Output 1:

5

Sample Input 2:

7

12 35 50 59 60 73 90

100

Sample Output 2:

-1

4. An array is **bitonic** if it consists of an increasing sequence of integers followed immediately by a decreasing sequence of integers. Given a bitonic array ***A*** of ***n*** distinct integers, write a program to check whether the given integer ***x*** is in the array (by modifying the **binary search)**. Assume that the array index starts from 0 and the array contains distinct elements. If ***x*** is present in ***A***, print the index of ***x*** in ***A***. Otherwise print -1.

   Input format:

   - The first line of the input contains an integer $n \in$ **[0, 10000],** the size of an array ***A***.
   - The second line contains ***n***, space-separated integers in the range **[−1000, 1000]**. These elements form the array ***A***.
   - The third line contains an integer $x \in$ **[−1000, 1000]** to be searched in the array.

   Output format:

   - If ***x*** is present in ***A***, print the index of ***x*** in ***A***.
   - If ***x*** is not present in ***A***, print −1.

   Sample Input 1:

   7

   12 35 59 90 73 60 50

   73

   Sample Output 1:

   4

   Sample Input 2:

   7

   12 35 59 90 73 60 50

   100

   Sample Output 2:

   -1

5. Write a program that contains a **main()** function and a recursive function **reverse().** The reverse() function should take as input a positive integer in the range [0, 1000000] and return an integer obtained by reversing the input integer. The main() function should read the input and print the output obtained from the reverse() function.

   Input format: Input is an integer ***n*** $\in$ **[0, 100000]** for which the reversed number is to be found.

   Output format: The output is the reversed number.

   Sample Input : 98765

   Sample Output: 56789

6. Write a program to print first ***n*** Fibonacci numbers using Recursion.

   Input format: Input is an integer ***n*** $\in$ **[0, 20]**.

   Output format: The output is a series of numbers separated by single space.

   Sample Input 1:
   2
   Sample Output 1:
   0 1
   Sample Input 2 :
   10
   Sample Output 2:
   0 1 1 2 3 5 8 13 21 34

****