Department of Computer Science and Engineering
Class Test - 1, Section - 16
Subject : Programming and Data Structure (CS19003)

Date: 31st December 2021 Time: 9:30 AM to 12:00 Noon Marks: 100

Instructions for :

- Give the name of the programs files as <Your roll>_<test number>_<question number>.c. For example, 21XX12345_T1_Q1.c for question 1 of test 1 of a student with roll 21XX12345. **Follow the file naming convention strictly**.
- Apart from the main .c file for each program, you should also upload one additional temporary .c file for each program (such as when you have finished half of the code). The naming for the temporary file should be in the format <Your roll>_<test number>_<question number>_temp.c. For example, 21XX12345_T1_Q1_temp.c **Make sure that your main code do not deviate much from its temporary code for each program**.
- You should upload the main .c file and the temporary .c file individually to the Moodle course web page once you finish answering each question. No need to zip the files.
- The **deadline** to upload the programs is 12:00 Noon. Beyond that, submission will be closed (No extensions will be granted).

---

1. Write a C program to perform the following:

   (a) Take two binary numbers (as long long int) of equal length as input. You have to check if the inputted numbers are binary or not, i.e each digit of both the numbers should be either 1 or 0. Also count the number of bits in both the numbers. Display an error message if any one of the numbers is not a binary number or the two numbers are not of the same bit length, and exit the program.

   (b) If both the numbers are valid binary numbers and they are of the same bit length, then, perform bitwise XOR operation of the two binary numbers from the least significant bit (i.e the rightmost bit) onwards. Display the bit by bit XOR output on the screen as you process each bit of the two numbers from right to left. You **cannot** use any bitwise operators available in the C library.

   NOTE:- The bitwise XOR output that you will display will actually be in the reverse format of the correct bitwise XOR. That is okay, do not worry. Also you do not need to generate any number or such for the bitwise XOR operation, simply print '1' or '0' side by side as per the correct operation. You can assume that the most significant bit of both the input numbers will be 1.

   [Bitwise XOR rules:
   0 XOR 0 = 0
   0 XOR 1 = 1
   1 XOR 0 = 1
   1 XOR 1 = 0] **[30 Marks]**

   Example 1:

Input:
10010
11001

Output:
11010 [Note that this is the reverse of the actual bitwise XOR]

Example 2:

Input:
10010
1101

Output:
Error

Example 3:

Input:
10230
11011

Output:
Error

2. Write a C program to read a sequence of positive integers to detect all possible non-decreasing subsequences and print the total number of such non-decreasing subsequences and the length of the longest non-decreasing subsequence. Within a series of integer numbers, a non-decreasing subsequence is a contiguous sequence of numbers where all are in non-decreasing order. There may be more than one non-decreasing subsequence within the entire sequence, your task is to print the total number of such subsequences present and the length of the longest such subsequence. You should continuously keep on taking inputs for the sequence until the user enters zero or a negative value. Your program must contain only one loop. Both scanning the next integer and processing the scanned integer should be done in that loop. Write no functions other than main(). Do not use any array.

   [**Hint:-** In order to get the length of longest subsequence, think of a counter variable and how it is incremented or decremented based on the new input and last input] **[30 Marks]**

   Example:

   Suppose you take input the following numbers one by one:-
   2, 4, 3, 5, 6, 1, 3, 4, 8, 9, 6, 7, 5, 3, 9, 0 [You stop taking input here]
   So the entire valid sequence is: {2, 4, 3, 5, 6, 1, 3, 4, 8, 9, 6, 7, 5, 3, 9}
   List of non-decreasing subsequences: {2, 4}, {3, 5, 6}, {1, 3, 4, 8, 9}, {6,7}, {5},{3, 9}
   Longest non-decreasing subsequence: {1, 3, 4, 8, 9}

   So, your output should be,
   Total number of non-decreasing subsequences: 6
   Length of the longest non-decreasing subsequence: 5

3. Write a C program to take input a number ‘n’ and print the following pattern as shown,

   For example,for n=3

```
*   *   *   *   *   *   *
*           *           *

*       *       *       *

*   *       *       *   *

*       *       *       *

*           *           *

*   *   *   *   *   *   *
```

   For n=4,

```
*   *   *   *   *   *   *   *   *

*               *               *

*           *       *           *

*       *               *       *

*   *           *           *   *

*       *               *       *

*           *       *           *

*               *               *

*   *   *   *   *   *   *   *   *
```

[40 Marks]