**Birla Institute of Technology & Science, Pilani**

**Second Semester 2022-23**

**CS F111 – Computer Programming**

**Online Programming Test**

**SET ORANGE**

==================================================================================

**25/06/2023 Max. Marks: 80M Duration: 150 mins**

==================================================================================

# **General Instructions**

* This paper consists of three questions.
* Read all the instructions and the problem statement very carefully before attempting.
* Carefully follow the submission instructions mentioned below before uploading your solution on the DomJudge portal.
* If you submit multiple submissions, only the latest one will be considered for evaluation. Whatever you submit on the DomJudge portal will be considered as final. **It is your responsibility to make sure that you are submitting the correct file.** Later, if some student claims that he/she has mistakenly submitted the wrong file, we won’t be entertaining any such request and we will evaluate based on whatever is submitted on the DomJudge portal.

# **Instructions to attempt the test**

- Create a new directory with your 13 digit ID number, e.g. if your ID is **2022A1PS0001P**, create a directory with this ID.

- Copy all the files extracted from this zip folder (except this PDF file) into this directory.

- Now, right click on the file “**myScript.sh**” which was just copied in the new directory and change the permissions/properties of the file to allow it to **Execute**. Do not change permissions of any other file that you copied.

- While attempting your questions, you are not allowed to modify the files myScript.sh, main.c, f1.h f2.h or f3.h, except giving execute permission to myScript.sh. You should neither make any changes to the function parameters, nor to the return types of any of the functions in this file

- You can only modify f1.c, f2.c or f3.c.

- Now start attempting the questions below.

- To compile and execute, you must use myScript.sh only, e.g., if you want to compile and test your solution for Q1, you must run the following on the terminal: **./myScript.sh 1**

- Similarly, to compile and test your solution for Q2, run **./myScript.sh 2**

-Similarly, to compile and test your solution for Q3, run **./myScript.sh 3**

*If you do any of the above dont’s you will definitely incur a heavy penalty.*

# **Submission Instructions**

Your final submission should be a zip folder with the name as your 13 digit ID number with 6 files in all (namely myScript.sh, main.c, f1.c, f1.h, f2.c, f2.h,f3.c,f3.h)

# **Problem Statements**

Q1. In the main function, you already have code to create array ***Arr*** of size ***n*** and store the user's input in it. After that we take an integer ***rotateBy*** from the user.

In the main function, we pass the array ***Arr, n*** *and* ***rotateBy*** in the function **rotateArray(Arr, n, rotateBy).**  Your task is to code the function **void rotateArray(int arr[], int n, int rotateBy)** in **f1.c** to rotate the **Arr** by a given number of positions, i.e. **rotateBy**.

The type of rotations depends on the integer given by the user, i.e., ***rotateBy***. If a given number is positive, then rotate the array in clockwise order by the given number. Otherwise, rotate the array in anti-clockwise order by the given number. Your code should satisfy the following sample.

**Sample 1:**

Enter the size of the array: 5

Enter the elements of the array: 2 3 5 1 4

Enter the number of positions to rotate (positive or negative):3

Rotated Array: 5 1 4 2 3

**Sample 2:**

Enter the size of the array: 5

Enter the elements of the array: 2 3 5 1 4

Enter the number of positions to rotate (positive or negative): 17

Rotated Array: 1 4 2 3 5

**Sample 3:**

Enter the size of the array: 5

Enter the elements of the array: 2 3 5 1 4

Enter the number of positions to rotate (positive or negative): -4

Rotated Array: 4 2 3 5 1

**Q2** In the main function, you already have code to prompt the user to enter the sizes of two linked lists, ***size1*** and ***size2***. Then, using two for loops, you take input from the user one by one to create the two linked lists using ***insertnodeEnd*** function. After the loops, you will have two fully created linked lists. Following that, you invoke the function ***printCommonElements***, passing the heads of the two linked lists as arguments. This function returns the head of a new linked lists which contains all the common elements present in both linked lists.

In the main function following function are called: **insertNodeEnd(head, data)** and **printCommonelements(head1, head2).** Your task is to codethe following functions **struct Node\* insertNodeEnd(struct Node\* head, int data)** and **struct node \* printCommonElements(struct Node\* head1, struct Node\* head2)** in **f2.c** to satisfy the following the sample.

**Sample 1:**

Enter the size of list 1 and list 2: 3 4

Enter the elements of list 1: 1 2 3

Enter the elements of list 2: 2 3 4 5

List 1: 1 2 3

List 2: 2 3 4 5

New Link list created of common elements: 2 3

**Sample 2:**

Enter the size of list 1 and list 2: 3 2

Enter the elements of list 1: 1 2 3

Enter the elements of list 2: 4 5

List 1: 1 2 3

List 2: 4 5

No common elements

**Sample 3:**

Enter the size of list 1 and list 2: 4 5

Enter the elements of list 1: 1 2 3 4

Enter the elements of list 2: 2 3 5 6 4

List 1: 1 2 3 4

List 2: 2 3 5 6 4

New Link list created of common elements:2 3 4

**Q3.** Your program should take an integer input from the user to determine the number of rows in the pattern and then display the pattern accordingly.

|  |  |  |
| --- | --- | --- |
| Enter the value of n: 5  \*  \*\*  \* \*  \* \*  \* \*  \* \*  \* \*  \*\*  \* | Enter the value of n: 6  \*  \*\*  \* \*  \* \*  \* \*  \* \*  \* \*  \* \*  \* \*  \*\*  \* | Enter the value of n: 8  \*  \*\*  \* \*  \* \*  \* \*  \* \*  \* \*  \* \*  \* \*  \* \*  \* \*  \* \*  \* \*  \*\*  \* |

In the main function, the following function is called **pattern(n).** Your task is to code the function **void pattern(int n)** in **f3.c**  to print the above patterns.