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

**First Semester 2022-23**

**CS F111 – Computer Programming**

**Online Programming Test**

**SET YELLOW**

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

**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 two arrays ***arr1*** and ***arr2*** of ***size1*** and ***size2***, respectively and store the user's input in both the arrays. The input will be given in sorted order for both the arrays.

In the main function, we pass both arrays and their sizes in the function **findMedian(arr1, size1, arr2, size2).** Your task is to code code the **int findMedian(int arr1[], int size1, int arr2[], int size2)** in **f1.c** which returns the median of the combined sorted array. For example if arr1=[1,4,5] and arr2=[2,3,6,7] then the median of combined sorted array [1,2,3,4,5,6,7] is 4.

Your code should satisfy the following sample.

**Sample 1:**

Enter the size of the first array: 3

Enter the elements of the first array (in sorted order): 0 1 6

Enter the size of the second array: 2

Enter the elements of the second array (in sorted order): 3 5

Median of the two arrays: 3

**Sample 2:**

Enter the size of the first array: 3

Enter the elements of the first array (in sorted order): 1 3 5

Enter the size of the second array: 5

Enter the elements of the second array (in sorted order): 2 4 6 8 10

Median of the two arrays: 4

**Sample 3:**

Enter the size of the first array: 3

Enter the elements of the first array (in sorted order): 1 2 3

Enter the size of the second array: 3

Enter the elements of the second array (in sorted order): 3 4 5

Median of the two arrays: 3

**Q2.** In the main function, you already have code to prompt the user to enter the size of linked lists, ***size***. Then, using for loop, you take input from the user one by one to create the linked list using **insertnodeEnd** function. After the loop, you will have fully created linked list. Following that, you invoke the function **reverseLinkedList**, passing the head of the linked lists as an argument. This function returns the reverse the linked lists and returns the head of the link list.

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

**Sample 1:**

Enter the size of the list: 5

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

Original list: 1 2 3 4 5

Reversed linked list: 5 4 3 2 1

**Sample 2:**

Enter the size of the list: 6

Enter the elements of the list: 2 4 6 8 1 3

Original list: 2 4 6 8 1 3

Reversed linked list: 3 1 8 6 4 2

**Sample 3:**

Enter the size of the list: 3

Enter the elements of the list: -1 -2 0

Original list: -1 -2 0

Reversed linked list: 0 -2 -1

**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.