

## Section 16

## Mid Semester Examination

PDS Lab

SET - II

11.09.2018

### Instructions:

Create a sub directory named as **MidSem**.

Give the name of the programs as <p>\_1.c, <p>\_2.c, .. etc. for the problem 1, 2....., respectively. Here <p> implies the part number. For example, Part-A

Store all the programs under this assignment in the directory **MidSem**.

Zip the entire directory **MidSem** and rename it as <R>\_ **MidSem** \_tar.gz. where <R> denotes your Roll No.

You should upload your zipped file to the Moodle course web page latest by 11:55 hrs.

### Part-A

1. Read the input values from the keyboard, calculate the following expression and then print the values on the display screen.

$$x = 1 - \frac{1}{3} + \frac{1}{3^2} - \frac{1}{3^3} + \dots (-1)^n \frac{1}{3^n}, \quad n \geq 0$$

(Do not use any loop to calculate the value of x.)

#### Test cases:

n = 5; n = -5; n = 1

2. Read a set of numbers (including both positive and negative values). Calculate the average of all positive numbers. Do not use any array.

#### Test cases:

-3 2 5 -5 6  
-9 -3 -7 -2 -1  
10 8 6 2 5  
55

3. Define an array say X of size 20 of integers. Read a number say n<20 from the user.

Read n numbers from the keyboard such that any  $i^{\text{th}}$  number is always less than all the numbers in  $X[0], X[1], \dots, X[i-1]$ . If the user enters any number, which violates the above condition, then you should report the same and ignore the entry of the number.

#### Test cases:

n = 4  
12 11 15 19 9 13 8

## Part-B

4. Read a string of characters and store it in an array say Y. Take care that the input string may contain any number of blank spaces. Print the string after reading. Arrange the string in the reverse ordering of their entries. You should not use any other array other than Y. Print the string after reversing it.

**Test cases:**

IIT Kharagpur

Welcome $\varphi$

Welcome to iit $\varphi\varphi\varphi$ Kharagpur (here  $\varphi$  denotes one blank space)

5. Read two sets of numbers from the user and then store them in two arrays say A and B. Find all the numbers, which are common in both the array A and B.

**Test cases:**

A: 3 5 7 9

A: 1 2 3 4

B: 1 3 5 7 11

B: 5 6 7

## Part-C

6. Define a recursive function which will return the greatest common division (GCD) between the largest and smallest numbers stored in an array.

Call this function from the `main()` for n times (the value of n should be read from the user). In each loop, you should read a set of non-negative numbers, find the smallest and largest numbers, call the recursive function and then print the result returned by the recursive function.

**Test cases:**

n = 4

1 2 3 4 6

100 10 50 0 5 15

7 11 13 19

---\*---

### Submission instruction

Zip your *MidSem* directory. Upload your zip file into the Moodle server.