Programming and Data Structures Laboratory | 2021-22 Autumn semester, Section 20
Lab Test 2 | February 01, 2022
Full marks: 3 x 20 = 60
Time: 09:15 – 11:45 am (2 hours 30 minutes)

Instructions

* There are three (3) problems in this test, each of 20 marks. The program for each problem must be written in an individual C source file. You should submit the following three plain text C source files: **<rollnumber>_T2_1.c, <rollnumber>_T2_2.c and <rollnumber>_T2_3.c** .

* Submissions must be through the course Moodle, before **11:45 AM** (according to the Moodle clock). If you miss this deadline, then you need to email your submission to TA Owais Iqbal (Email: owais.iqbal@kgpian.iitkgp.ac.in) within 12:00 noon; there will be a penalty of 30 % of marks for such late submissions. No submission will be allowed after 12:00 noon; any submission reaching the TA's mailbox after 12:00 (according to the receipt timestamp of the email) will be rejected.
* You are NOT supposed to take the help of any person/TA/book/online material during the test. Any malpractice/plagiarism will be penalised severely, with the minimum being awarding zero for the entire test.
* It is your responsibility to make your programs understandable, through meaningful variable names, indentation, comments (if necessary). Programs that are not understandable will be penalized.

---

1. [20 Marks] Define a structure: `struct point{float x, y;}`; Write a function that takes two points as input and returns their Euclidian distance to the calling function. Its prototype will be: `float calculate_distance(struct point p, struct point q).`
Declare an array of such points in `main()`. You can take the maximum size of the array as 10. From the user take `n` as the number of points to be entered, fill up the array with `n` points. Find the minimum distance among these `n` points by calling the function `calculate_distance` from `main()` and print this distance.

Example output:

```
Enter no. of points (2 to 10): 3
Enter (x,y) coordinates of the points:
0 0
3 3
2 2
Minimum inter-point distance = 1.414214
```

```
Enter no. of points (2 to 10): 5
Enter (x,y) coordinates of the points:
11.11 22.22
22.22 33.33
33.33 44.44
44.44 55.55
55.55 66.66
Minimum inter-point distance = 15.711910
```

2. [20 Marks] Write a function that takes as parameters two numbers x (real) and y (positive integer) and returns the value of x + x^2 + … + x^y. Call this function from `main()` and print the value of the expression **from `main()`**. Note the examples below. You should print both the series expression as well as the value of the series expression. You are **not allowed** to use `math.h` header file.

Example output:

Enter x: 4
Enter y: 3
4.000000 + 4.000000^2 + 4.000000^3 is: 84.000000

Enter x: 1.5
Enter y: 4
1.500000 + 1.500000^2 + 1.500000^3 + 1.500000^4 is: 12.187500

3. [20 Marks] Write a C program that should define a **square** matrix in the main function. In the main function call the following functions to perform some matrix operations. The maximum number of rows or columns can be taken as 4.

a. [3 marks] **void matrixInput(float mat[][N])**: This function should fill the 2D array with user supplied values by appropriately prompting the user. Display the array nicely formatted (i.e., in row-column format).

b. [7 Marks] **void matTranspose (float A[][N], float AT[][N])**: This function should first display the matrix A nicely formatted. Then it should transpose the matrix A and store the transposed matrix in AT and display the transposed matrix.

c. [10 Marks] **void OrthoNormal(float A[][N], float AT[][N])**: This function should first display the matrix A nicely formatted. A square N X N matrix "A" is said to be orthonormal if A*AT = I where AT is the transpose of the matrix A and I is the NXN identity matrix. Determine whether the matrix A is orthonormal or not and display an appropriate message.

Example of an Orthonormal Matrix is:
0.00 -0.80 -0.60
0.80 -0.36 0.48
0.60 0.48 -0.64

Example of a matrix which is not Orthonormal is:
1.00 2.00
3.00 4.00