

## Foundations of Technical Programming

### Week 7

You may be asked to demonstrate/explain your work to the tutor, if you are absent/unavailable or fail to demonstrate properly, zero marks will be awarded.

#### Submission Requirement

1. Source code
2. Screen shot of the terminal which shows the output.

The following sample program reads 10 values into a 2D array of 2 rows & 5 columns. The content of the first 2D array is then copied to another 2D array.

```
#include <stdio.h>

int main()
{
    int A1[2][5], B1[2][5], i, j;

    printf("Enter 10 values into the first array ...\n");
    for(i=0;i<2;i++)
    {
        for(j=0;j<5;j++)
        {
            scanf("%d",&A1[i][j]);
        }
    }

    printf("Copying the values from first array to the second array \n");
    for(i=0;i<2;i++)
    {
        for(j=0;j<5;j++)
        {
            B1[i][j] = A1[i][j];
        }
    }
    printf("The values of the second arrays ...\n");
    for(i=0;i<2;i++)
    {
        for(j=0;j<5;j++)
        {
            printf("%d",B1[i][j]);
        }
    }

    return 0;
}
```

## Task 7.1

Using C programming language, write a program that fills random values (0-100) into two 2D arrays of 3 rows and 3 columns.

Add the values of both the arrays and store that into the 3<sup>rd</sup> 2D array of same size. Print the values of the 3<sup>rd</sup> 2D array.

Example:

Array A

4	5	2
0	2	2
4	9	2

Array B

22	99	21
10	34	97
4	34	23

Array C

26	104	23
10	36	99
8	43	25

## Task 7.2

A two dimensional array of size 10 rows and 4 columns, contains the marks of 10 students in 4 subjects. And a single dimensional array for size 10 has been created to store the average of the 10 students.

Write a C program, that will calculate the average of 4 marks and store the average in the single dimensional array.

Example:

19	36	66	55	44
88	65	66	59	69.5

## Task 7.3

Create a single dimension integer array of size 20.

Fill the array with 10 random numbers between 1 and 100 in alternate position of the array starting from the first position.

Fill the remaining 10 empty position of the array with the previous positions value \* 2.

Print the array from 1st to the last value.

```

Sara@DESKTOP-5I2SSJH /c/COS10008
$ out1
38    76
47    94
34    68
40    80
92    184
78    156
61    122
35    70
24    48
1     2
  
```