CSE115L – Programming Language I Lab

Lab – 21

2-D Arrays

In this lab, we will solve a few problems using arrays. The following examples will help you remember the syntax.

Basic syntax for 2-dimensional array

DataType ArrayName [number of rows][number of columns];

Example: `int myArray[3][4];`

myArray	0	1	2	3
0				
1				
2				

Initialization:

```
int myArray[3][4] = {
   {0, 1, 2, 3} ,   /*  row  0 */
   {4, 5, 6, 7} ,   /*  row  1 */
   {8, 9, 10, 11}   /*  row  2 */
};
```

myArray	0	1	2	3
0	0	1	2	3
1	4	5	6	7
2	8	9	10	11

To access value at row index 0 and column index 1, we write a[0][1]

Example 1: Assigning and printing elements in a 2-dimensional array

```
int main()
{
    int A[100][100], i, j, rows, columns;
    printf("Number of rows: ");
    scanf("%d",&rows);
    printf("Number of columns: ");
    scanf("%d",&columns);

    for(i=0;i<rows;i++)
    {
        for(j=0;j<columns;j++)
        {
            printf("A[%d][%d]: ",i, j);
            scanf("%d",&A[i][j]);
        }
    }
    printf("Values in array A:\n");

    for(i=0;i<rows;i++)
    {
        for(j=0;j<columns;j++)
        {
            printf("%10d ",A[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

Perform the following tasks.

Task 1: i) Declare a 2-dimensional array of row size 3 and column size 3.
ii) Fill the 2-dimensional array with values from the user.
iii) Search for a user given value in the 2D array. If the value is not present, then print "Not found", otherwise print "Found"

Task 2: In this task, you will find the summation of the elements in a specific column of a matrix. Read two integers, **m** and **n**, from the user. These are the dimensions of the matrix. Next, read the elements of the matrix from the user. Then, read which column you have the sum. Finally, print the sum.

Sample Run:

Input			Output
Enter number of rows: 3 Enter number of columns: 4	Enter matrix elements: 2 3 6 7 4 8 1 5 9 0 7 2	Enter which column to find sum of: 2	Sum of column 2: 11