Multidimensional Arrays

ADVANCED COLLEGE
OF ENGINEERING & MANAGEMENT
Affiliated to Tribhuvan University (T.U.)

Introduction

- Multi-Dimensional arrays are those which have more than one dimension.
- Are defined in same manner as one dimensional array, except a separate square bracket is required for each dimension or index
- The Two Dimensional array is also called matrix.

Declaration & Initialization of two-dimensional array

- Just like 1-D arrays, it can be declared as:

  data_type array_name [row_size] [column_size];

  - **Example:**

    int matrix [2] [3];

    float m [10] [20];

- Initialization:
  - **Example:**

    int marks [2] [3]={ {2,4,6},{8,10,12} };

- is equivalent to: marks[0] [0]=2; marks [0][1]=4; marks[0][2]=6;
  marks[1][0]=8; marks[1][1]=4; marks[1][2]=12;

Accessing 2-D array elements

- A two – dimensional array can be seen as a table with ‘x’ rows and ‘y’ columns
- Where, the row number ranges from 0 to (x-1) and,
- column number ranges from 0 to (y-1).
- A two – dimensional array ‘x’ with 3 rows and 3 columns is shown below:

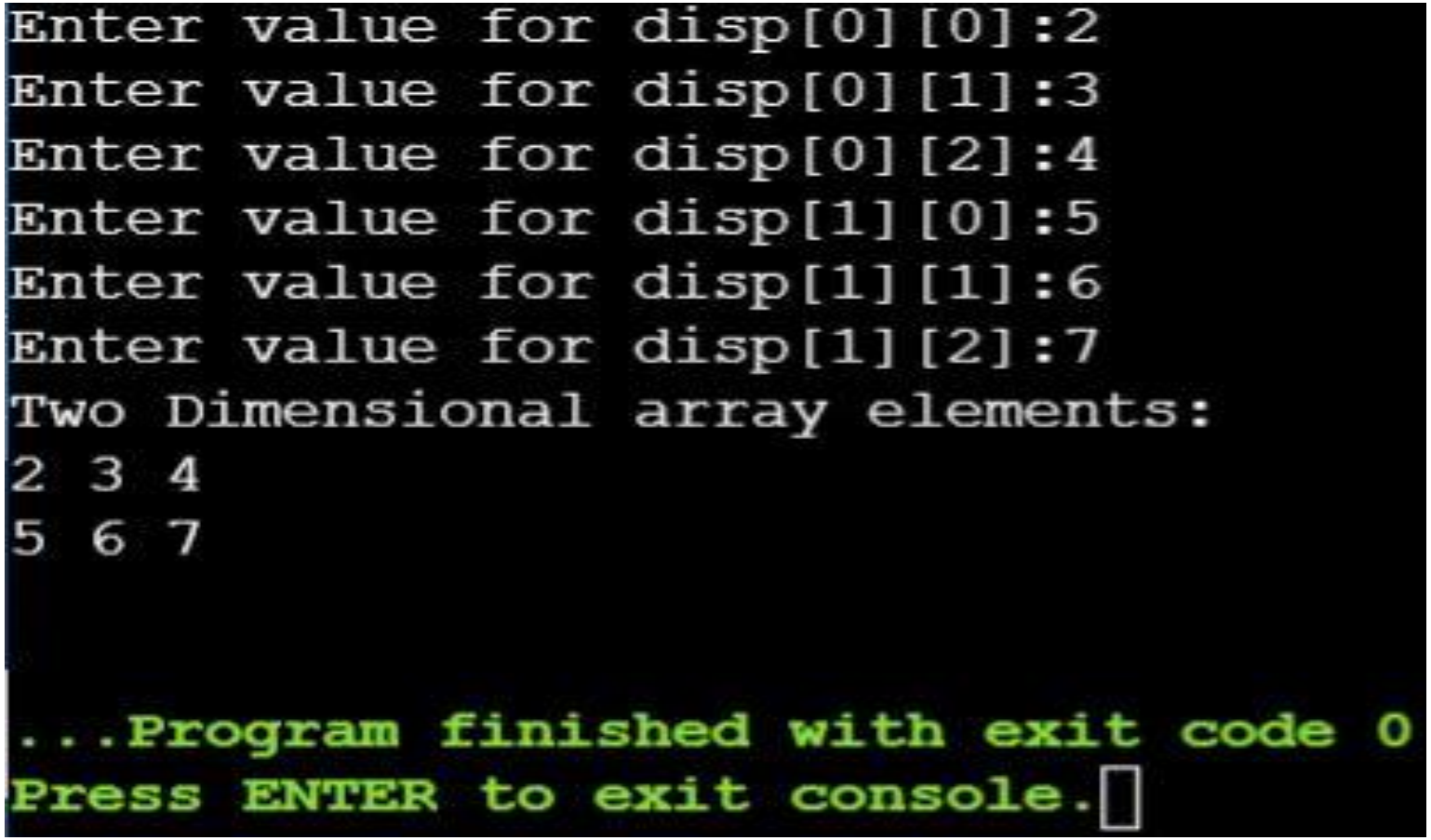
	Column 0	Column 1	Column 2
Row 0	x[0][0]	x[0][1]	x[0][2]
Row 1	x[1][0]	x[1][1]	x[1][2]
Row 2	x[2][0]	x[2][1]	x[2][2]

C-Program Example

```
int main(){
   /* 2D array declaration*/
   int disp[2][3];
   /*Counter variables for the loop*/
   int i, j;
   for(i=0; i<2; i++) {
      for(j=0;j<3;j++) {
         printf("Enter value for disp[%d][%d]:", i, j);
         scanf("%d", &disp[i][j]);
      }
   }
```

```
//Displaying array elements
   printf("Two Dimensional array elements:\n");
   for(i=0; i<2; i++) {
      for(j=0;j<3;j++) {
         printf("%d ", disp[i][j]);
         if(j==2){
            printf("\n");
         }
      }
   }
   return 0;
}
```

Output


Enter value for disp[0][0]:2
Enter value for disp[0][1]:3
Enter value for disp[0][2]:4
Enter value for disp[1][0]:5
Enter value for disp[1][1]:6
Enter value for disp[1][2]:7
Two Dimensional array elements:
2 3 4
5 6 7
...Program finished with exit code 0
Press ENTER to exit console.

Classwork:

- WAP to read a matrix of order *m*n* from user and multiply each element of the matrix by 3.

Solution

```
#include<stdio.h>
int main(){
    int m,n;
    printf("Enter m and n for the matrix: ");
    scanf("%d%d",&m,&n);
   /* 2D array declaration*/
   int disp[m][n];
   /*Counter variables for the loop*/
   int i, j;
   for(i=0; i<m; i++) {
     for(j=0;j<n;j++) {
        printf("Enter value for disp[%d][%d]:", i, j);
        scanf("%d", &disp[i][j]);
     }
   }
```

Solution

```
// Multiplying each term of matrix by 3
    for(i=0;i<m;i++){
        for(j=0;j<n;j++){
            disp[i][j]=3*disp[i][j];
        }
    }

   //Displaying array elements
   printf("Two Resultant Matrix is:\n");
   for(i=0; i<m; i++) {
      for(j=0;j<n;j++) {
         printf("%d\t", disp[i][j]);
         }
         printf("\n");
      }
   return 0;
}
```

Passing Arrays to Function

- Like ordinary variables and values, it is possible to pass the value of an array element and even an entire array as argument to a function.
- Syntax for function call passing array as argument:

  function_name(array_name);

- Syntax for function prototype which accepts array as argument:

  return_type function_name (data_type array_name[]);

Example: passing array element as an argument

```
#include <stdio.h>
void display(int age1, int age2)
{
   printf("%d\n", age1);
   printf("%d\n", age2);
}

int main()
{
   int ageArray[] = {2, 8, 4, 12};

   // Passing second and third elements to display()
   display(ageArray[1], ageArray[2]);
   return 0;
}
```

Example: Passing Arrays to Function

```
// Program to calculate the sum of array elements by passing to a function

#include <stdio.h>
float calculateSum(float []);

int main() {
   float result, age[6] = {23.4, 55, 22.6, 3, 40.5, 18};

   // age array is passed to calculateSum()
   result = calculateSum(age);
   printf("Result = %.2f", result);
   return 0;
}
```

Example: Passing Arrays to Function

```
float calculateSum(float age[]) {

  float sum = 0.0;

  for (int i = 0; i < 6; ++i) {
                    sum += age[i];
  }

  return sum;
}
```

Classwork:

- WAP to read ***n*** numbers in an array. Pass this array to a function which finds and displays the sum of even numbers only and the product of odd numbers only.

```
#include<stdio.h>
#include<conio.h>
void calculate(int[],int);
int main(){
int nums[50],n,i;
printf("How many numbers in the array? ");
 scanf("%d",&n);
 printf("Enter the Numbers: ");
 for(i=0;i<n;i++)
   {       scanf("%d",&nums[i]);
 }
 calculate(nums,n);
 getch();
  return 0;
}
```

```
void calculate(int ns[], int n)
{    int sum=0,product=1,i;
for(i=0;i<n;i++)
{
if (ns[i]%2==0)
{
sum+=ns[i];
  }
  else{
   product*=ns[i];
   }
 }
 printf("The sum of even numbers: %d",sum);
 printf("\nThe product of odd numbers:%d",product);
}
```