UNIT III – ARRAYS AND STRINGS IN C

Arrays: Initialization - One dimensional, Two dimensional, and Multi-dimensional arrays - String: Basics, declaring and initializing strings, string handling functions: standard and user defined functions.

3.2 TYPES OF ARRAYS IN C

There are two types of arrays based on the number of dimensions it has. They are as follows:

1. One Dimensional Arrays (1D Array)
2. Multidimensional Arrays

ONE DIMENSIONAL ARRAYS IN C

The One-dimensional arrays, also known as 1-D arrays in C are those arrays that have only one dimension.

Syntax of 1D Array in C

```
array_name [size];
```

Example of 1D Array in C

```
// C Program to illustrate the use of 1D array
#include <stdio.h>
int main()
{
    // 1d array declaration
    int arr[5];
    // 1d array initialization using for loop
    for (int i = 0; i < 5; i++) {
        arr[i] = i * i - 2 * i + 1;
    }
    printf("Elements of Array: ");
    // printing 1d array by traversing using for loop
    for (int i = 0; i < 5; i++) {
        printf("%d ", arr[i]);
    }
    return 0;
}
```

Output

```
Elements of Array: 1 0 1 4 9
```

Array of Characters (Strings)

In C, we store the words, i.e., a sequence of characters in the form of an array of characters terminated by a NULL character. These are called strings in C language.

```
// C Program to illustrate strings
#include <stdio.h>
int main()
{
    // creating array of character
```

```
    char arr[6] = { H', 'e', 'l', 'l', 'o', '\0' };
    // printing string
    int i = 0;
    while (arr[i]) {
        printf("%c", arr[i++]);
    }
    return 0;
}
```

Output

```
Hello
```

Example Programs

1. Program to calculate sum of elements in an Array

Program:

```
#include<stdio.h>
#include<conio.h
void main()
{
int arr[100], size, i, sum = 0;
printf("Enter array size\n");
scanf("%d",&n);
printf("Enter array elements\n");
for(i = 0; i < n; i++)
scanf("%d", &arr[i]);
            //add all elements to the variable sum.
for(i = 0; i < n; i++)
{sum = sum + arr[i];
}
printf("Sum of the array = %d\n",sum);      }}
```

2. C Program to perform Linear search

Program

```
#include <stdio.h>
 int linearSearch (int arr[], int n, int target)
{
   int i;
   for (i = 0; i< n; i++)
{
       if (arr[i] == target)
{
           return i; }
   }
   return -1;
 }
  int main() {
   int arr[] = {10, 2, 8, 5, 17};
   int n = sizeof(arr)
 printf(“enter the element to be searched”);
 scanf(“%d”, &target);
   int result = linearSearch(arr, n, target);
   if (result == -1)
{
printf("Element not found in the array.\n");
   }
 else
{
printf("Element found at index: %d\n", result);
   }
   return 0;
}
```

Multidimensional Array in C

Multi-dimensional Arrays in C are those arrays that have more than one dimension. Some of the popular multidimensional arrays are 2D arrays and 3D arrays. We can declare arrays with more dimensions than 3d arrays but they are avoided as they get very complex and occupy a large amount of space.

3.1 TWO-DIMENSIONAL ARRAY IN C

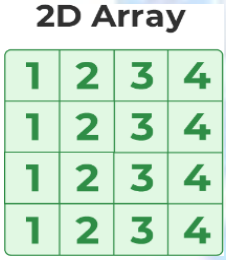
A Two-Dimensional array or 2D array in C is an array that has exactly two dimensions. They can be visualized in the form of rows and columns organized in a two-dimensional plane.

Syntax of 2D Array in C

```
array_name[size1] [size2];
```

Here,

- **size1:** Size of the first dimension.
- **size2:** Size of the second dimension.



Example of 2D Array in C

1. C Program to illustrate 2d array

```
#include <stdio.h>
int main()
{
   // declaring and initializing 2d array
   int arr[2][3] = { 10, 20, 30, 40, 50, 60 };
  printf("2D Array:\n");
   // printing 2d array
   for (int i = 0; i < 2; i++) {
      for (int j = 0; j < 3; j++) {
         printf("%d ",arr[i][j]);
```

```
        }
        printf("\n");
    }

    return 0;
}
```

Output

```
2D Array:

10 20 30

40 50 60
```

2. Program for Matrix Addition

Program:

```
#include <stdio.h>
int main() {
    int m, n, i, j;
    printf("Enter the number of rows and columns of the matrices: ");
    scanf("%d%d", &m, &n);
    int a[m][n], b[m][n], c[m][n];
    printf("Enter the elements of matrix A: \n");
    for (i = 0; i < m; i++) {
        for (j = 0; j < n; j++) {
            scanf("%d", &a[i][j]);
        }
    }
    printf("Enter the elements of matrix B: \n");
    for (i = 0; i < m; i++) {
        for (j = 0; j < n; j++) {
            scanf("%d", &b[i][j]);
        }
    }
```

```
  // add the matrices
  for (i = 0; i < m; i++) {
    for (j = 0; j < n; j++) {
      c[i][j] = a[i][j] + b[i][j];
    }
  }
  // print the result
  printf("The sum of the two matrices is: \n");
  for (i = 0; i < m; i++) {
    for (j = 0; j < n; j++) {
      printf("%d ", c[i][j]);
    }
    printf("\n");
  }
  return 0;
}
```

Output

```
Enter the number of rows and columns of the matrices: 2 2
Enter the elements of matrix A:
1 2
3 4
Enter the elements of matrix B:
5 6
7 8
The sum of the two matrices is:
6 8
10 12
```

3. 3. Write a program to perform Matrix Multiplication

Program:

```
#include<stdio.h>
#include<stdlib.h>
```

```
int main(){
int a[10][10],b[10][10],mul[10][10],r,c,i,j,k;
system("cls");
printf("enter the number of row=");
scanf("%d",&r);
printf("enter the number of column=");
scanf("%d",&c);
printf("enter the first matrix element=\n");
for(i=0;i<r;i++)
{
for(j=0;j<c;j++)
{
scanf("%d",&a[i][j]);
}
}
printf("enter the second matrix element=\n");
for(i=0;i<r;i++)
{
for(j=0;j<c;j++)
{
scanf("%d",&b[i][j]);
}
}
 printf("multiply of the matrix=\n");
for(i=0;i<r;i++)
{
for(j=0;j<c;j++)
{
mul[i][j]=0;
for(k=0;k<c;k++)
{
mul[i][j]+=a[i][k]*b[k][j];
```

```
} } }
//for printing result
for(i=0;i<r;i++)
{
for(j=0;j<c;j++)
{
printf("%d\t",mul[i][j]);
}
printf("\n");
}
return 0;
}
```

Output:

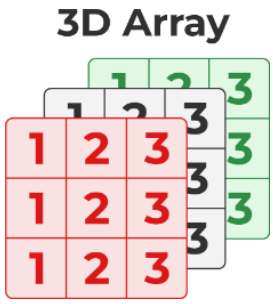
```
enter the number of row=3
enter the number of column=3
enter the first matrix element=
1 1 1
2 2 2
3 3 3
enter the second matrix element=
1 1 1
2 2 2
3 3 3
multiply of the matrix=
6 6 6
12 12 12
18 18 18
```

3.2 THREE-DIMENSIONAL ARRAY IN C (MULTI-DIMENSIONAL ARRAY)

Another popular form of a multi-dimensional array is Three Dimensional Array or 3D Array. A 3D array has exactly three dimensions. It can be visualized as a collection of 2D arrays stacked on top of each other to create the third dimension.

Syntax of 3D Array in C

```
array_name [size1] [size2] [size3];
```



Example of 3D Array

```
// C Program to illustrate the 3d array
#include <stdio.h>
int main()
{
    // 3D array declaration
    int arr[2][2][2] = { 10, 20, 30, 40, 50, 60 };
    // printing elements
    for (int i = 0; i < 2; i++) {
        for (int j = 0; j < 2; j++) {
            for (int k = 0; k < 2; k++) {
                printf("%d ", arr[i][j][k]);
            }
            printf("\n");
        }
        printf("\n \n");
    }
    return 0;
}
```

Output

```
10 20
30 40
50 60
0 0
```