

ARRAYS

Introduction:

An array is a fixed-sized sequenced collection of elements of the same data type. It is simply a grouping of like – type data. In its simplest form, an array can be used to represent a list of numbers, or a list of names.

An array is a sequenced collection of related data items that share a common name.

Different types of Arrays:

- One –dimensional arrays
- Two - dimensional arrays
- Multidimensional arrays

One – dimensional arrays:

A list of items can be given one variable name using only one subscript and such a variable is called a single-subscripted variable or a one – dimensional array.

Declaration of One – Dimensional Arrays :

Syntax:

type variable – name [size];

Example:

```
int a[10];
```

Program :

```
#include<stdio.h>
void main()
{
    int a[3], i;
    for(i=0 ; i<3;i++)
    {
        scanf("%d",&a[i]);
        printf("The value of i is %d",a[i]);
    }
    getch();
}
```

Output :

```
1 1 2
```

```
The value of i is 0
```

```
The value of i is 1
```

```
The value of i is 2
```

Initialization of One – Dimensional Arrays:

An array is declared, its elements must be initialized. Otherwise, they will contain “garbage” . An array can be initialized at either of the following stages.

- At compile time

- At run time

Compile Time Initialization:

We can initialize the elements of arrays in the same way as the ordinary variables when they are declared.

Syntax:

Type array-name [size] = {list of values};

Example:

```
int i[3]={0,0,0};
```

Program :

```
#include<stdio.h>
void main()
{
    int a[3]={ 0, 0,1};
    clrscr ();
    printf("The value of a is %d",a[0]);
    printf("The value of a is %d",a[1]);
    printf("The value of a is %d",a[2]);
    getch();
}
```

Output :

```
The value of a is 0
The value of a is 1
The value of a is 2
```

Two – Dimensional Arrays:

C allows us to define tables of items by using two – dimensional arrays.

Syntax:

Type array_name[row_size][column_size];

Example:

```
int a[2][2] = {0,0,1,1};
```

Program :

```
#include<stdio.h>
void main()
{
    int a[3][3], i, j, k;
    clrscr();
    for( i = 0 ; i < 3 ; i ++ )
    {
        for( j = 0 ; j < 3 ; j ++ )
        {
            k = i * j;
        }
    }
}
```

```

        }
        printf(" The value of k is %d\n",k);
    }
    getch();
}

```

Output :

```

The value of k is 0
The value of k is 0
The value of k is 0
The value of k is 0
The value of k is 1
The value of k is 2
The value of k is 0
The value of k is 2
The value of k is 4

```

Multi Dimensional Arrays:

C allows arrays of three or more dimensions. The exact limit is determined by the compiler.

Syntax:

Type array_name[s1][s2][s3].....[sm];

Example:

```
int a[2][2][2];
```

Program :

```

#include<stdio.h>
void main()
{
    int i , j , k , l , a[3][3][3];
    clrscr();
    for(i=0;i<3;i++)
    {
        for( j=0 ;j<3;j++)
        {
            for( k=0 ; k<3; k++)
            {
                l = i * j * k;
                printf("The value of l is %d\n",l);
            }
            printf(" The j loop is going to be executed\n");
        }
    }
}

```

```
        }
        printf("The i loop is going to be executed\n");
    }
    getch();
}
```

Output :

```
The value of I is 0
The value of I is 0
The value of I is 0
The j loop is going to be executed
The value of I is 0
The value of I is 0
The value of I is 0
The j loop is going to be executed
The value of I is 0
The value of I is 0
The value of I is 0
The j loop is going to be executed
The I loop is going to be executed
The value of I is 0
The value of I is 0
The value of I is 0
.....
.....
.....
```