

Unit - 3

1D Array in C

A one-dimensional (1D) array is a collection of elements of the same data type stored in contiguous memory locations. Each element is accessed using an index.

Declaration of 1D Array

Array declaration specifies the data type, array name, and number of elements.

Syntax: data_type array_name[size];

Example: int marks[5];

This declares an integer array named marks that can store five elements.

Array indexing:

Element	Index
First element	0
Second element	1
Third element	2
Fourth element	3
Fifth element	4

In C, array indexing starts from 0 and ends at size - 1.

Input in 1D Array

The scanf() function is used to take input for array elements.

Example:

```
int a[5], i;
for (i = 0; i < 5; i++)
{
    scanf("%d", &a[i]);
}
```

The loop takes input for all five elements, one by one.

Output of 1D Array

The printf() function is used to display array elements.

Example:

```
int a[5] = {10, 20, 30, 40, 50};  
int i;  
for (i = 0; i < 5; i++)  
{  
    printf("%d ", a[i]);  
}
```

Output: 10 20 30 40 50

Traversal of 1D Array

Array traversal is the process of visiting and accessing each element of an array one by one.

A for loop is commonly used for traversal.

Syntax:

```
for (i = 0; i < size; i++)  
{  
    // Process array element  
}
```

Example:

```
#include <stdio.h>  
  
int main()  
{  
    int a[5] = {10, 20, 30, 40, 50};  
    int i;  
    for (i = 0; i < 5; i++)  
    {  
        printf("a[%d] = %d\n", i, a[i]);  
    }  
    return 0;  
}
```

Output:

a[0] = 10

a[1] = 20

a[2] = 30

a[3] = 40

a[4] = 50

Complete Program: Input, Output and Traversal

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

```
int main()
```

```
{
```

```
    int a[5], i;
```

```
    printf("Enter 5 elements:\n");
```

```
    for (i = 0; i < 5; i++)
```

```
    {
```

```
        scanf("%d", &a[i]);
```

```
    }
```

```
    printf("Array elements are:\n");
```

```
    for (i = 0; i < 5; i++)
```

```
    {
```

```
        printf("%d ", a[i]);
```

```
    }
```

```
    return 0;
```

```
}
```

Input:

10 20 30 40 50

Output:

Array elements are:

10 20 30 40 50