Session 06 B

Wednesday, 8 October 2025 9:08 AM

Array :

In C programming, an **array** is a collection of elements (all of the same data type) stored in **contiguous memory locations**. Arrays allow you to store and access multiple values using a single variable name and an index.

🔷 Syntax of an Array

type arrayName[size];

- type: the data type of elements (e.g., int, float, char)
- arrayName: the name of the array
- size: the number of elements in the array

🔷 Example: Integer Array

```
#include <stdio.h>
int main() {
   int numbers[5] = {10, 20, 30, 40, 50};
// Accessing elements
   printf("First element: %d\n", numbers[0]);
   printf("Third element: %d\n", numbers[2]);
// Modifying elements
   numbers[1] = 25;
// Loop through the array
   for(int i = 0; i < 5; i++) {
       printf("Element %d: %d\n", i, numbers[i]);
   }
return 0;
}
```

🔷 Types of Arrays in C

1. **One-dimensional array** – Basic form of array (as above)
2. **Two-dimensional array** – Used like a table or matrix

Example: 2D Array

```
int matrix[2][3] = {
   {1, 2, 3},
   {4, 5, 6}
};
```

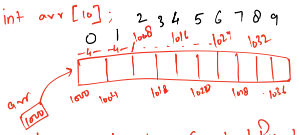
3. **Multidimensional arrays** – Arrays with more than two dimensions (less common in basic programs)

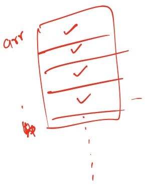
🔷 Notes

- Array indices start from 0
- The size of the array must be known at compile time (unless using **dynamic memory allocation**)
- Out-of-bounds access (e.g., arr[10] in a size-5 array) leads to **undefined behavior**

Declaration   datatype NameofArray [size];   int - 4 Byte (1000)

int arr [10];

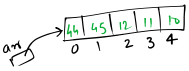




↳ Array Name is a Constant Pointer which Holds Address

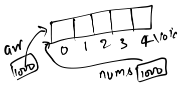
The value Can't Be changed.

int arr [5];



arr[0] = 44;
arr[1] = 45;
arr [2] = 12;
arr [3] = 11
arr [4] = 10

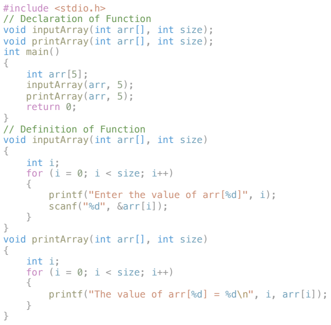
```
# include <Stdio.h>
int main() {
    int arr [5];
    inputArray ( arr, 5 ) ;
 → outputArray (       ) ;
}
```



```
void inputArray ( int nums [], int n )
{
for( int i = 0 ; i < n ; i++)
  { printf(" Enter the value of index [%d] ", i);
    Scanf(" %d", &nums[i]);
  }
}
```

```
void outputArray ( int arr[], int n )
  {
   Printf(" The Array value are Given By \n ) ;
   printf(" [ " );
   for( int i = 0; i < n; i++)
    {
       printf(" %d   ", arr[i]);
    }
   printf(" ]\n");
  }
```

[_ _ _ _]



```
#include <stdio.h>
// Declaration of Function
void inputArray(int arr[], int size);
void printArray(int arr[], int size);
int main()
{
    int arr[5];
    inputArray(arr, 5);
    printArray(arr, 5);
    return 0;
}
// Definition of Function
void inputArray(int arr[], int size)
{
    int i;
    for (i = 0; i < size; i++)
    {
        printf("Enter the value of arr[%d]", i);
        scanf("%d", &arr[i]);
    }
}
void printArray(int arr[], int size)
{
    int i;
    for (i = 0; i < size; i++)
    {
        printf("The value of arr[%d] = %d\n", i, arr[i]);
    }
}
```

int arr [4] = { 2, 4, 9, 11 };

int arr [ ] = { 1, 2, 4, 9, 3 };

nums [3, 0, 1]
[1, 2, 3]

③

XOR = 0; xoR = XOR ^ 3 ^ 1

```
Missing (int* nums, int n)
{
        int xoR = 0;
    for( int i=0 ; i< n ; i++)
        {   xoR= xoR ^ nums[i] ^ (i+1);

        }
        return xoR;
        }
}
```

xoR = 0 ^ 3 ^ 1

3^1 = 011
001
0|0

xoR= xoR ^ 0 ^ 2;

0 ~~2~~ ^ 0 ^ ~~2~~

② xoR = 0 ^ 3^1

011 =②
001

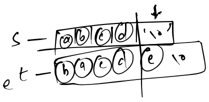
268, 136

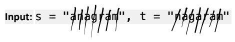
4 1 7 1 2

```
char xoR='\0' ;
for( i=0; s!='\0' ; i++)
 {
  xoR= xoR ^ s[i] ^ t[i]
 }
    xoR= xoR ^ t[i] ;
        return xoR;
```

abcd

abcde




Input: s = "anagram", t = "nagaram"

Start

L 00000000 . - - - - - - -1010

0111

Goal

111