Course Name:	PROGRAMMING IN C	Semester:	II
Date of Performance:	03-02-2025	DIV/ Batch No:	[redacted]
Student Name:	[redacted]	Roll No:	[redacted]

Experiment No: 4

Title: Write a program in C to demonstrate use of arrays

Aim and Objective of the Experiment:

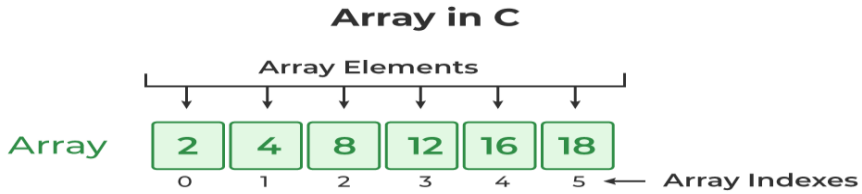
1.Write a program in C to read n number of values in an array and display them in reverse order.
2. Program to sort the 1D array in the ascending or descending order
3. Write a program in C to count the total number of duplicate elements in an array.

COs to be achieved:

CO3: Apply the concept of arrays and string

Theory:

An array in C is a fixed-size collection of similar data items stored in contiguous memory locations. It can be used to store the collection of primitive data types such as int, char, float, etc., and also derived and user-defined data types such as pointers, structures, etc.

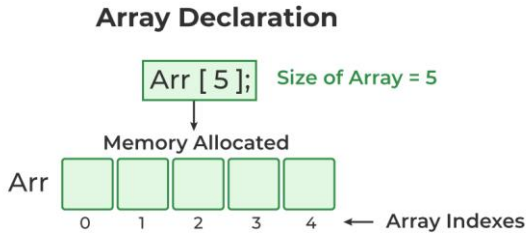


C Array Declaration

In C, we have to declare the array like any other variable before using it. We can declare an array by specifying its name, the type of its elements, and the size of its dimensions. When we declare an array in C, the compiler allocates the memory block of the specified size to the array name.

Syntax of Array Declaration

data_type array_name [size];

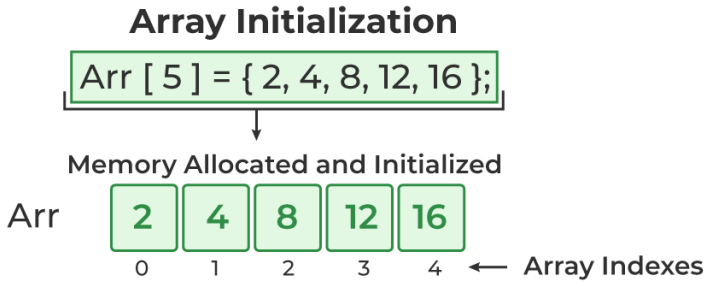



```
// C Program to illustrate the array declaration
#include <stdio.h>

int main()
{

   // declaring array of integers
   int arr_int[5];
   // declaring array of characters
   char arr_char[5];

   return 0;
}
```



Array Initialization with Declaration without Size

```
data_type array_name[] = {1,2,3,4,5};
```

Array Initialization after Declaration (Using Loops)

```
for (int i = 0; i < N; i++)
    {
       array_name[i] = valuei;
    }
```

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

- **One Dimensional Arrays (1D Array)**
  Syntax of 1D Array in C
  array_name [size];

- **Two-Dimensional Array in C**
  array_name[size1] [size2];

Problem Statements:

1. Write a program in C to read n number of values in an array and display them in reverse order.
2. Program to sort the 1D array in the ascending or descending order
3. Write a program in C to count the total number of duplicate elements in an array.

Code :

Question 1.]

```
#include <stdio.h>
int main()
{
   int n, i;
   printf("Enter the number of elements: ");
   scanf("%d", &n);
   int arr[n];
   printf("Enter %d elements:\n", n);
   for(i = 0; i < n; i++)
   {
    scanf("%d", &arr[i]);
   }
    printf("Elements in reverse order:\n");
   for(i = n - 1; i >= 0; i--)
   {
    printf("%d ", arr[i]);
   }
   return 0;
}
```

Question 2.]

```
#include <stdio.h>
int main()
{
   int arr[1000];
   int size;
   int i, j, temp;
   printf("Enter size of array: ");
   scanf("%d", &size);
   printf("Enter elements in array: ");
   for(i=0; i<size; i++)
   {
      scanf("%d", &arr[i]);
   }
   for(i=0; i<size; i++)
   {
      for(j=i+1; j<size; j++)
      {
         if(arr[i] > arr[j])
         {
            temp     = arr[i];
            arr[i] = arr[j];
            arr[j] = temp;
         }
      }
   }
   printf("\nElements of array in ascending order: ");
   for(i=0; i<size; i++)
   {
      printf("%d\t", arr[i]);
   }
   return 0;
}
```

Question 3.]

```
#include <stdio.h>
int main()
{
   int a[10],i,j,n,c=0 ;
   printf("Enter size of the array : ");
   scanf("%d", &n);
   printf("\nEnter elements in array : \n ");
   for(i=0; i<n; i++)
   {
      printf("\nelement - %d :", i);
      scanf("%d",&a[i]);
   }

   for(i=0; i<n; i++)
   {
      if(a[i]!=-1)
      {
         for(j=i+1; j<n; j++)
         {
            if(a[i]==a[j])
            {
               c++;
               a[j]=-1;
            }
         }
       }
   }
   printf("duplicate numbers in the  array: %d", c);
   return 0;
}
```

Output:

Question 1.]

```
Enter the number of elements: 5
Enter 5 elements:
1 2 3 4 5
Elements in reverse order:
5 4 3 2 1
Process returned 0 (0x0)   execution time : 6.016 s
Press any key to continue.
```

Question 2.]

```
Enter size of array: 5
Enter elements in array: 4 70 80 20 45

Elements of array in ascending order: 4 20      45      70      80
Process returned 0 (0x0)   execution time : 14.498 s
Press any key to continue.
```

Question 3.]

```
Enter size of the array : 5

Enter elements in array :

element - 0 :2

element - 1 :5

element - 2 :6

element - 3 :7

element - 4 :8
duplicate numbers in the  array: 0
Process returned 0 (0x0)   execution time : 11.887 s
Press any key to continue.
```

Post Lab Subjective/Objective type Questions:

1. **Write a program in C to find the maximum and minimum elements in an array.**

Ans. *CODE*

```
#include <stdio.h>
int main(){
  int array[100];
  int i, max, min, size;
  printf("Enter size of the array: ");
  scanf("%d", &size);
  printf("Enter elements in the array: ");
  for(i=0; i<size; i++){
     scanf("%d", &array[i]);
  }
  max = array[0];
  min = array[0];
  for(i=1; i<size; i++){
     if(array[i] > max){
        max = array[i];

     }
     if(array[i] < min)
      {
        min = array[i];
```

```
            }
        }
        printf("Maximum element = %d \n", max );
        printf("Minimum element = %d \n", min );
        return 0;
    }
```

OUTPUT

```
Enter size of the array: 5
Enter elements in the array: 0 56 87 97 100
Maximum element = 100
Minimum element = 0

Process returned 0 (0x0)   execution time : 11.639 s
Press any key to continue.
```

2. Write a program in C for adding two matrices of the same size.

Ans. *CODE*

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

      }
   }
   printf ("Enter elements of matrix B: \n");
   for (i=0; i<m; i++)
   {
      for (j=0; j<n; j++)
      {
         scanf ("%d" , &B[i][j]);
      }
   }
   printf ("Sum of matrices: \n");
   for (i=0 ; i<m ; i++)
```

```
    {
        for (j=0 ; j<n ; j++)
        {
            sum[i][j]= A[i][j] + B[i][j];
            printf ("%d " , sum[i][j]);
        }
        printf ("\n");
    }
    return 0;
}
```

OUTPUT

```
Enter rows and columns: 2
2
Enter elements of matrix A:
1
2
3
4
Enter elements of matrix B:
1
2
3
4
Sum of matrices:
2 4
6 8

Process returned 0 (0x0)   execution time : 5.776 s
Press any key to continue.
```

Conclusion:

In this module we learnt about array, declaration of array and types of arrays.
An array in C is a fixed-size collection of similar data items stored in contiguous memory locations. It can be used to store the collection of primitive data types such as int, char, float, etc., and also derived and user-defined data types such as pointers, structures, etc.
Array Declaration: **data_type array_name [size];**

Signature of faculty in-charge with Date: