**C Array**

An array is defined as the collection of similar type of data items stored at contiguous memory locations. Arrays are the derived data type in C programming language which can store the primitive type of data such as int, char, double, float, etc. It also has the capability to store the collection of derived data types, such as pointers, structure, etc. The array is the simplest data structure where each data element can be randomly accessed by using its index number.

C array is beneficial if you have to store similar elements. For example, if we want to store the marks of a student in 6 subjects, then we don't need to define different variables for the marks in the different subject. Instead of that, we can define an array which can store the marks in each subject at the contiguous memory locations.

By using the array, we can access the elements easily. Only a few lines of code are required to access the elements of the array.

**Declaration of an Array**

**type variable\_name[];**

In the above example, type denotes the data type of the array and variable\_name denotes the name of the array.

**Examples of some basic Array of specific data type**

int a[10];

float a[20];

double a[20];

char a[20];

**How to declare and Initialize an Array in C**

int a[5]={8,56,34,57,78};

int a[]= {3,5,6,8,9};

float a[]={13.5,23.7,45.12,-34.4,-12.8};

char a1[7]="Sumit";

**Some Important points about array by taking the above example**

* All array elements would be present in contiguous memory Locations.
* 4 bytes for the integer variable. If an array has 6 elements then it consumes total of 24 bytes of memory.
* Arrays indexing always starts from a[0].
* In the above Example of a C array, each array occupies indexes from a[0] to a[5].

**In addition, below We have also mentioned some properties of an array. So please have a look over it.**

**Properties of An Array in C Program**

**An Array has the following properties**

* Elements of an array should be of a similar data type.
* It takes memory a contiguous fashion.
* Array elements are stored in a continuous fashion so that they can be randomly accessed.

**Now together with properties let’s have an eye on the advantages and disadvantages of an Array.**

**Advantage of C Array in C Program**

1) Code Optimization: Data can be accessed with very less code.

2) Ease of traversing: Traversing of an Array is very easy with for loop.

3) Ease of sorting: We can easily sort Array elements.

4) Random Access: Also we can access elements randomly.

5) Data Structure Implementation: Array helps to implement various data structures like linked lists, stacks, queues, trees, graphs, etc.

6) Matrix Representation: With the help of 2 Array, Matrix can be represented.

**Disadvantage of C Array**

1). Time complexity: In insertion and deletion operations, Time complexity increases.

2). Memory Wastage: Because arrays are fixed in size so if no elements in an array, still it consumes all spaces.

3). Fixed In Size: Once you had declared the size of an array, it is not possible to increase it.

**Types of Array in C**

**There are 2 types of C arrays**

1. **One dimensional array**
2. **Multi dimensional array**
   * Two dimensional array
   * Three dimensional array

**One Dimensional Array**

This can be considered as array as a row, where elements are stored one after another.

Syntax

**data-type arr\_name[array\_size];**

**data-type:**It denotes the type of the elements in the array.

**arr\_name**: Name of the array. It must be a valid identifier.

**array\_size**: Number of elements an array can hold.

**Example 1: 1D Array implementation In C**

#include <stdio.h>

int main()

{

int a=0;

int marks[4];

marks[0]=10;

marks[1]=20;

marks[2]=30;

marks[3]=40;

marks[4]=50;

for(a=0;a<5;a++){

printf("%d \n",marks[a]);

}

return 0;

}

Output

10

20

30

40

50

**Example 2: 2-D Array implementation In C**

**#include <stdio.h>**

**int main()**

**{**

**int i,j;**

**int arr[2][2] = {10,20,30,40};**

**for (i=0;i<2;i++)**

**{**

**for (j=0;j<2;j++)**

**{**

**printf("value of arr[%d] [%d] : %d\n",i,j,arr[i][j]);**

**}**

**}**

**return 0;**

**}**

**Output**

**value of arr[0] [0] : 10**

**value of arr[0] [1] : 20**

**value of arr[1] [0] : 30**

**value of arr[1] [1] : 40**

**Example 3: 2-D Array implementation In C**

**#include <stdio.h>**

**int main()**

**{**

**int i,j,arr[2][2];**

**arr[0][0] = 10;**

**arr[0][1] = 20;**

**arr[1][0] = 30;**

**arr[1][1] = 40;**

**for (i=0;i<2;i++)**

**{**

**for (j=0;j<2;j++)**

**{**

**printf("value of arr[%d] [%d] : %d\n",i,j,arr[i][j]);**

**}**

**}**

**return 0;**

**}**

**Output**

**value of arr[0] [0] : 10**

**value of arr[0] [1] : 20**

**value of arr[1] [0] : 30**

**value of arr[1] [1] : 40**

**Write a program to calculate the sum of 50 numbers entered by the user**

#include<stdio.h>

#include<conio.h>

void main()

{

clrscr();

int sum = 0;

for(int i=0;i<50;i++)

{

sum += arr[i];

}

printf("%d", sum);

getch();

}

Program to compute the average of first 200 numbers

#include<stdio.h>

#include<conio.h>

void main()

{

clrscr();

float avg= 0.0;

for(inti=0; i<200; i++)

{

avg += arr[i];

}

avg = avg/200;

printf("%d", avg);

getch();

}

**Write a program to find the maximum element in an array**

#include<stdio.h>

#include<conio.h>

void main()

{

clrscr();

int maxel = arr[0];

for (int i=1; i<100; i++)

{

if (arr[i]>maxel)

{

maxel=arr[i];

}

}

printf("%d", maxel);

getch();

**Program to store 10 numbers in an array**

//take input of 10 numbers in array and print them

#include<stdio.h>

#include<conio.h>

void main()

{

clrscr();

int i, a[10];

printf("Enter the 10 number of the array\n");

for(i=0;i<10;i++)

{

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

}

printf("The elements of the Array are\n");

for(i=0;i<10;i++)

{

printf("A[%d]=%d\n",i,a[i]);

}

getch();

}

**Next, write a program to store 10 numbers in an array and print them in the reverse order**

//take the input of 10 numbers from user and print in reverse order

#include<stdio.h>

#include<conio.h>

void main()

{

clrscr();

int a[10],i;

printf("enter the 10 numbers to be printed in reverse\n");

for(i=0;i<10;i++)

{

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

}

printf("The reverse order of the numbers are\n");

for(i=9;i>=0;i--)

{

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

}

getch();

}

**Write a program in C to find the largest and smallest element in an array**

//Take an array from the user and find the largest and smallest

#include<stdio.h>

#include<conio.h>

void main()

{

clrscr();

int i,greatest,lowest,a[10];

printf("Enter the elements of an array\n");

for(i=0;i<10;i++)

{

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

}

greatest = a[0];

for(i=1;i<10;i++)

{

if(greatest>a[i])

greatest = greatest;

else

greatest = a[i];

}

printf("Greatest=%d",greatest);

lowest = a[0];

for(i=1;i<10;i++)

{

if(lowest<a[i])

lowest=lowest;

else

lowest=a[i];

}

printf("\nLowest=%d",lowest);

getch();

}

**Write a program using C to arrange the elements of an array in an ascending order**

//take in 10 elements of an array and rearrange the array in increasing order

#include<stdio.h>

#include<conio.h>

void main()

{

clrscr();

int i,a[10],temp;

printf("Enter the elements of the array\n");

for(i=0;i<10;i++)

{

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

}

for(int j=0;j<10;j++)

{

for(i=0;i<10;i++)

{

if(a[i]<a[i+1])

{

a[i]=a[i];

}

else

{

temp=a[i];

a[i]=a[i+1];

a[i+1]=temp;

}

}

}

printf("The array arranged in increasing order is\n");

for(i=0;i<10;i++)

{

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

}

getch();

}

**Addition of two matrix in C**

C program for matrix addition:

#include <stdio.h>  
   
int main()  
{  
   int m, n, c, d, first[10][10], second[10][10], sum[10][10];  
   
   printf("Enter the number of rows and columns of matrix**\n**");  
   scanf("%d%d", &m, &n);  
   printf("Enter the elements of first matrix**\n**");  
   
   for (c = 0; c < m; c++)  
      for (d = 0; d < n; d++)  
         scanf("%d", &first[c][d]);  
   
   printf("Enter the elements of second matrix**\n**");  
   
   for (c = 0; c < m; c++)  
      for (d = 0 ; d < n; d++)  
         scanf("%d", &second[c][d]);  
     
   printf("Sum of entered matrices:-**\n**");  
     
   for (c = 0; c < m; c++) {  
      for (d = 0 ; d < n; d++) {  
         sum[c][d] = first[c][d] + second[c][d];  
         printf("%d**\t**", sum[c][d]);  
      }  
      printf("**\n**");  
   }  
   
   return 0;  
}

**C program to find transpose of a matrix**

#include <stdio.h>  
   
int main()  
{  
  int m, n, c, d, matrix[10][10], transpose[10][10];  
   
  printf("Enter the number of rows and columns of a matrix**\n**");  
  scanf("%d%d", &m, &n);

  printf("Enter elements of the matrix**\n**");  
   
  for (c = 0; c < m; c++)  
    for (d = 0; d < n; d++)  
      scanf("%d", &matrix[c][d]);  
   
  for (c = 0; c < m; c++)  
    for (d = 0; d < n; d++)  
      transpose[d][c] = matrix[c][d];  
   
  printf("Transpose of the matrix:**\n**");  
   
  for (c = 0; c < n; c++) {  
    for (d = 0; d < m; d++)  
      printf("%d**\t**", transpose[c][d]);  
    printf("**\n**");  
  }

  return 0;  
}

|  |
| --- |
|  |