What is the array?

Arrays are simply a collection of similar data types stored at contiguous memory locations. It can store primitive types of data like int, char, float, double etc. With the help of the arrays, a programmer can access the elements very easily. Suppose, you want to store marks of 20 students then you might try to declare 20 variables like student1_marks, student2_marks etc. But what if i tell you that you can do all those with a single variable named array. By the help of a few lines of code, you can access those elements.

Properties of array in C

- An array is a variable that can store a fixed-size collection of elements of the same data type.
- You can access the elements inside of an array randomly. You can also calculate the address of each element in an array.
- Elements of the array stored at contiguous memory locations.

Advantages of an array in C

- You can access the elements of the array randomly.
- You can also sort the elements in an array with just a few lines of code.
- It is easy to access the data of the array.

Disadvantages of an array in C

- The array has to be fixed size. You cannot exceed the limit of it. And it does not have the feature to grow dynamically like LinkedList

How to declare an array?

Syntax for declaring an array:-

```
dataType array_name[arraySize];
```

Declare an array

```
int student_marks[20];
char student_name[10];
float numbers[5];
```

Note:- After the declaration, size of the array cannot be changed.

Access elements of an array in C

In C, you can access the elements of the array using their indices.

Suppose, you declare an array like below:-

```
int numbers[5];
```

If you want to access the first element of the above array then do **numbers[0]** and if you want to access the second element then do **numbers[1]** and so on.

- The first index of the array is 0 not 1. That's why **numbers[0]** is the first element.
- To access the last element of the array, do **numbers[4].**

Initialize an array:-

At the time of declaration, it is possible to initialize an array.

Syntax to initialize an array:-

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

You can also initialize an array like below:-

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

In case, if you don't specify the size of the array during initialization then the compiler will know it automatically.

numbers[0] = 2
numbers[1] = 3
numbers[2] = 4
numbers[3] = 5
numbers[4] = 6

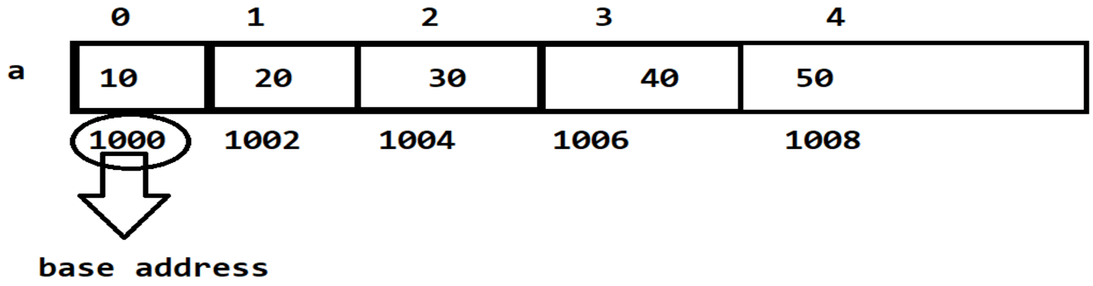
Basic Example of an array:-

```
#include<stdio.h>
int main()
{
int numbers[5] = {2, 3, 4, 5, 6};
printf("TechVidvan Tutorial: Basic Example of an array!");
printf("First number is : %d\n",numbers[0]);
printf("Third number is : %d\n",numbers[2]);
printf("Last number is : %d\n",numbers[4]);
}
```

Q. what is the base Address of Array ?

0th position address called as base address in array means starting position of address called as base address shown in following diagram.

```
int a[5]={10,20,30,40,50};
```



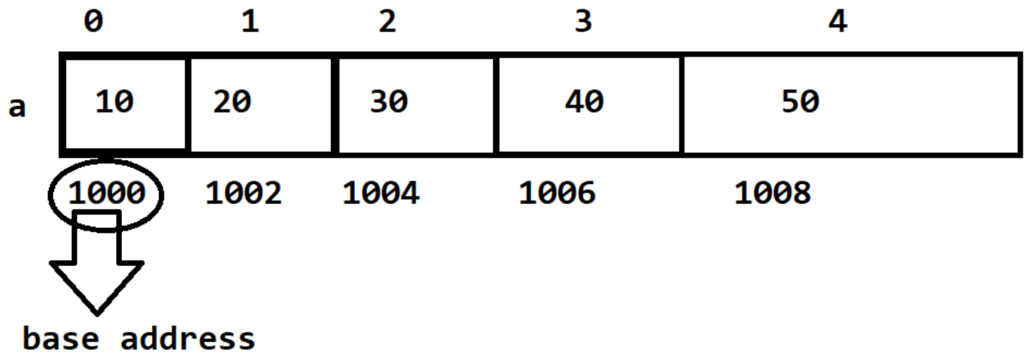
Q. how to get the base address of array in c language ?

If we want to get the address of array we have the two ways

a) Access the array without subscript:

Example given below:

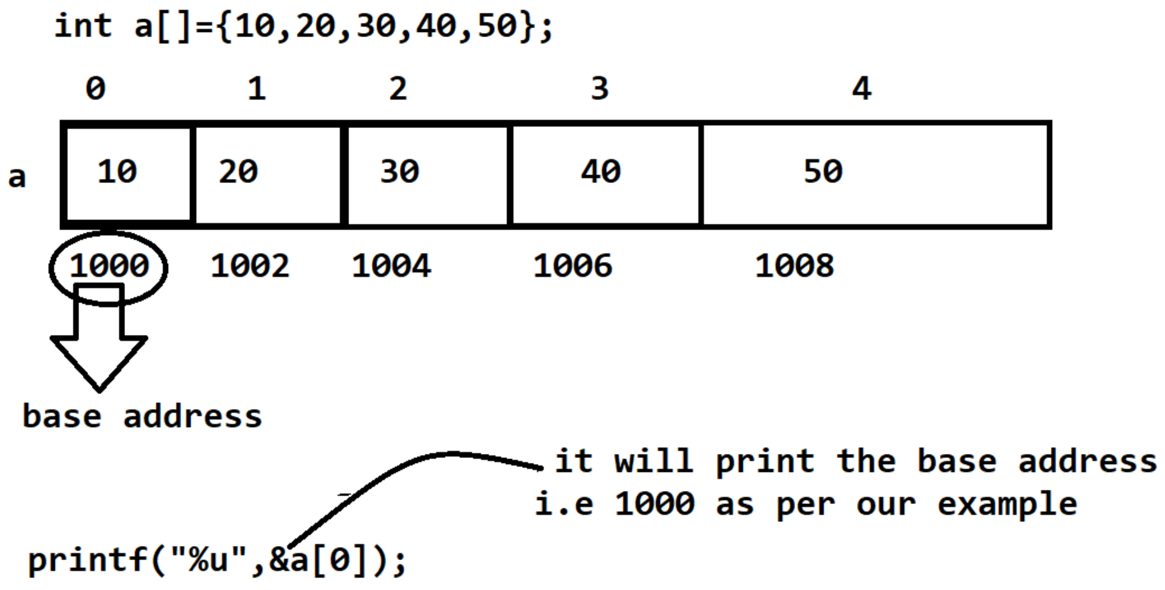
```
int a[]={10,20,30,40,50};
```



```
printf("%u",a);
```

it will print the base address
i.e 1000 as per our example

b) Use the array with 0th index and & operator



```
int a[]={10,20,30,40,50};
```

it will print the base address i.e 1000 as per our example

```
printf("%u",&a[0]);
```

Q. Explain the types of array?

There are two types of array in programming

1) Single dimensional array

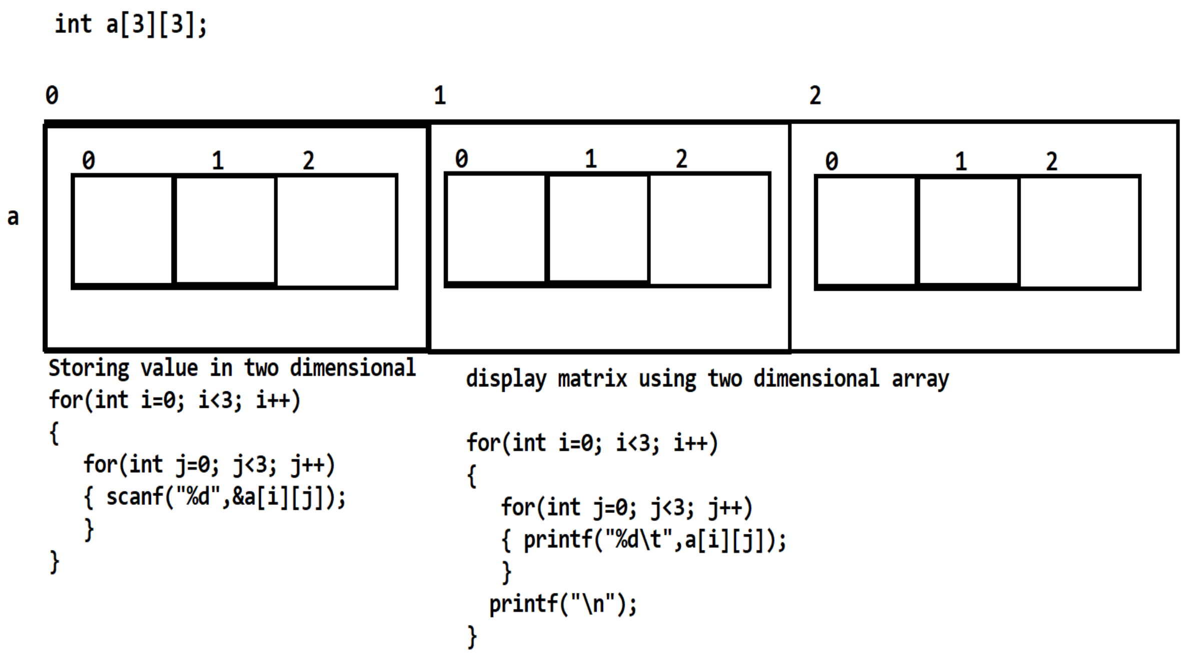
2) Multi dimensional array

Q. what is the two dimensional array and why use it?

Two dimensional array is used for create the matrix in programming but it is not a matrix in programming it is just sequentially memory representation and every block having a two digit index matrix creation from two dimensional array is just a logic of programmer.

Example: int a[3][3];

The following diagram indicate how to two dimensional array store in memory



```
int a[3][3];
```

Storing value in two dimensional

```
for(int i=0; i<3; i++)
{
   for(int j=0; j<3; j++)
   { scanf("%d",&a[i][j]);
   }
}
```

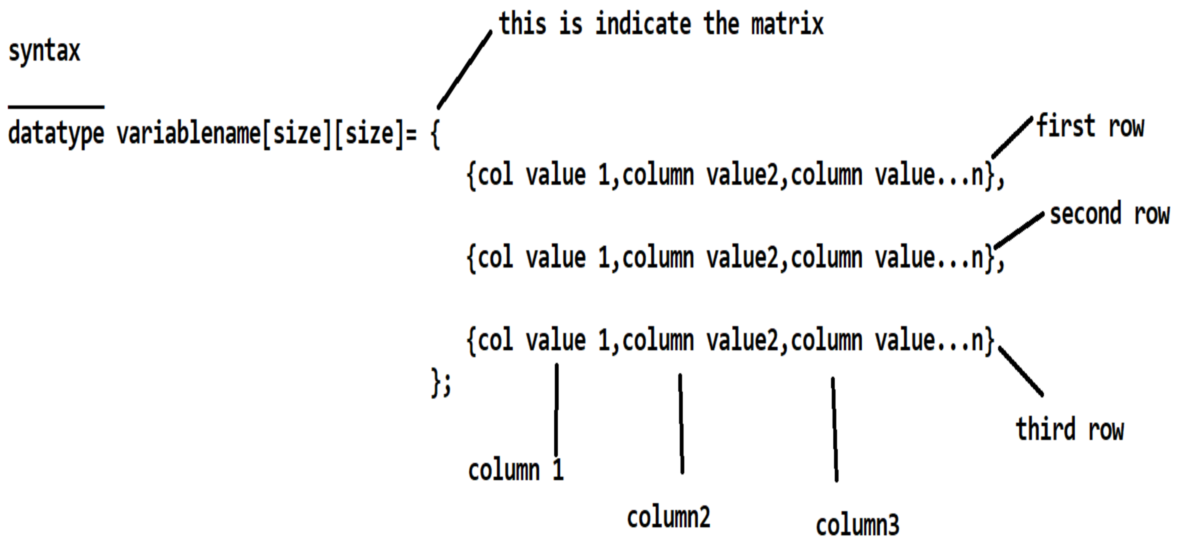
display matrix using two dimensional array

```
for(int i=0; i<3; i++)
{
   for(int j=0; j<3; j++)
   { printf("%d\t",a[i][j]);
   }
  printf("\n");
}
```

Q. how we can initialize the two dimensional array?

Array initialization means we can fix the value in array at the time of array declaration.

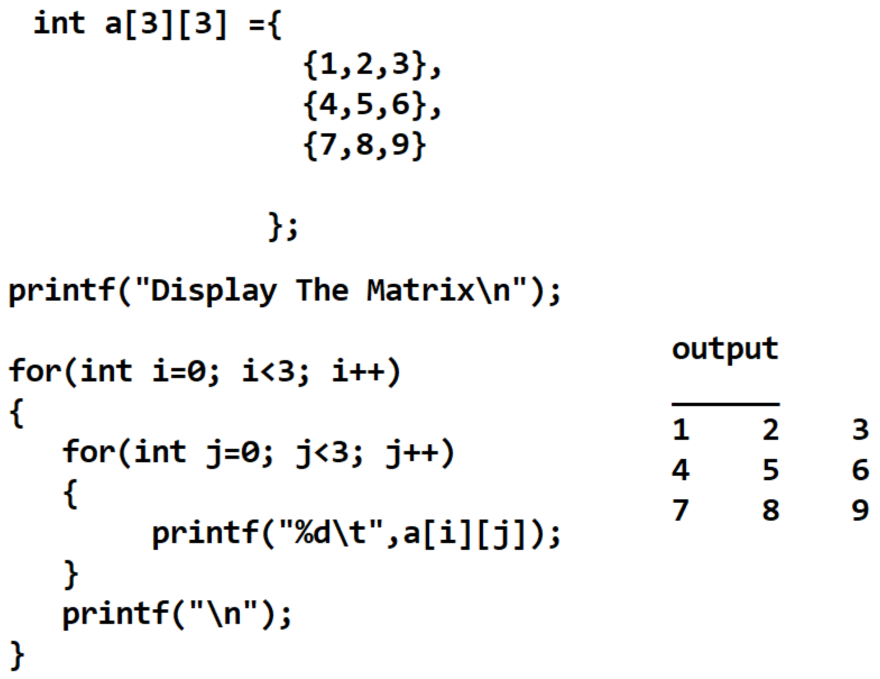
Following Example shows the array initialization of two dimensional array



syntax

```
datatype variablename[size][size]= {
        {col value 1,column value2,column value...n},
        {col value 1,column value2,column value...n},
        {col value 1,column value2,column value...n}
};
```

Example:



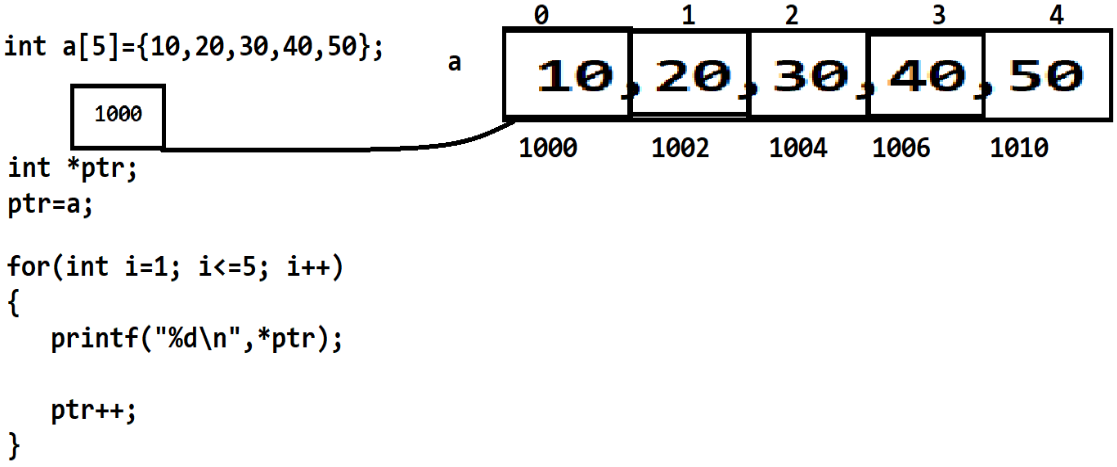
```
  int a[3][3] ={
                 {1,2,3},
                 {4,5,6},
                 {7,8,9}

               };
printf("Display The Matrix\n");

for(int i=0; i<3; i++)
{
   for(int j=0; j<3; j++)
   {
        printf("%d\t",a[i][j]);
   }
   printf("\n");
}
```

```
output
______
1     2     3
4     5     6
7     8     9
```

Q. How to access the array by using pointer?

If we want to access the array by using pointer we have to store the base address of array in pointer and means your pointer can point the array and we can access the array by using a pointer.

Following Example and Diagram shows the above concept.



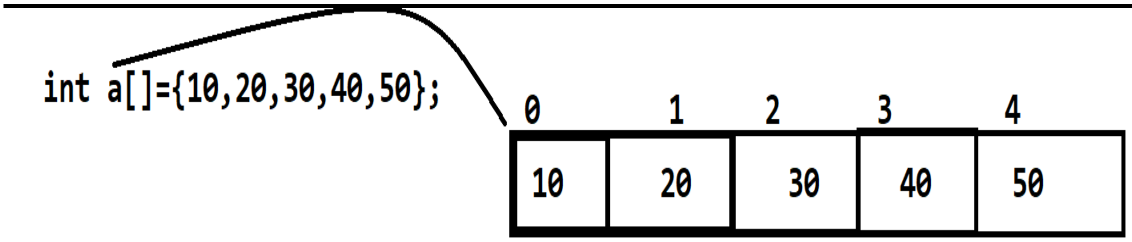
```
int a[5]={10,20,30,40,50};

int *ptr;
ptr=a;

for(int i=1; i<=5; i++)
{
   printf("%d\n",*ptr);

   ptr++;
}
```

In Above Example we store the base address of array in pointer means our pointer ptr points to base address i.e 1000 and when for loop get executed then in for loop i=1 i<=5; i++ means first i value is 1 and 1 less than equal to 5 so condition is true then control enter in for loop and display the value on 1000 address i.e 10 as our example then ptr++ statement get executed means our pointer move on next block i.e 1002 and then again i++ get executed then the value of i is 2 so 2<=5 this condition again get true so control enter in loop and display the value on 1002 address and again ptr++ get executed. this statement get executed 5 times and so on.

Q. Can you modify the base address of array?

No we cannot modify or change the base address of array if we try to modify the base address of array in c language then we get error LValue required.

Following Example shows the above concept



```
int a[]={10,20,30,40,50};
```

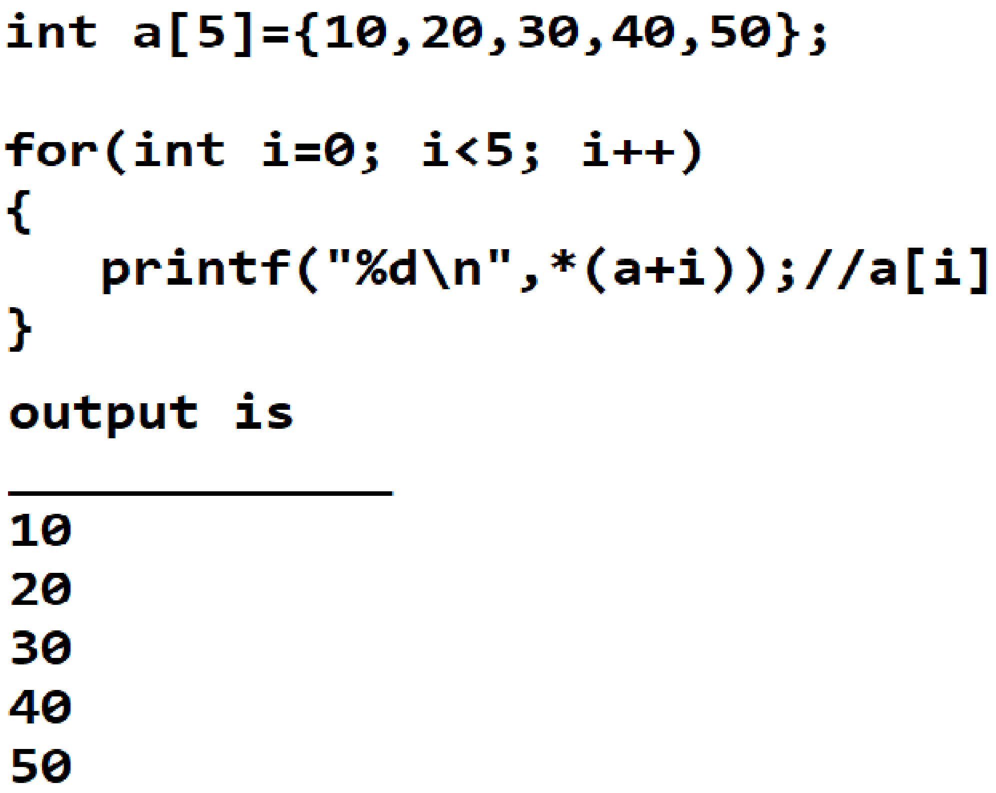
0	1	2	3	4
10	20	30	40	50

```
printf("%u",a++); //we try to modify the base address

Error : LValue Required
```

Q. What is the meaning of a[i]?

a[i] indicate the base address + index internally it is equal with *(a+i)

Following Example shows the meaning of a[i]?



```
int a[5]={10,20,30,40,50};

for(int i=0; i<5; i++)
{
   printf("%d\n",*(a+i));//a[i]
}
output is
_______________
10
20
30
40
50
```

The following statements are equal means base address + index

a[i]=i[a]=*(a+i)=*(i+a)=*(&a[i]) all statements are used for display the output of array.

MCQ Question on Array

Q.1 what will be the output of given code?

```
#include<iostream>
using namespace std;
int main(){
        int j=1;
        int a[5]={10,20,30,40,50};
        int c=a[j++];
        printf("C =%d\n",c);
```

```
	return 0;
}
```

Q.2 what will be the output of given code?

```
#include<iostream>
using namespace std;
int main(){
	int j=1;
	int a[5]={10,20,30,40,50};
	int c=a[++j];
	printf("C =%d\n",c);
	return 0;
}
```

Q.3 what will be the output of given code?

```
#include<iostream>
using namespace std;
int main(){
	int j=1;
	int a[5]={5,4,3,5,2};
	int d=a[4];
	int e=a[d]+10;
	int f=a[d+2];
	printf("D=%d\tE=%d\tF=%d\n",d,e,f);
	return 0;
}
```

Q.5 what will be the output of given code?

____

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

Q.6 what will be the output of given code?

____

```
#include<iostream>
using namespace std;
int main(){
  int a[5]={10,20,30,40,50};
  int i=1;
   for(;i!=5;){
            printf("%d\n",a[i++]);
          }
        return 0;
}
```

Q.7 what will be the output of given code?

____

```
#include<iostream>
using namespace std;
```

```
int main (){
  int a[5]={10,20,30,40,50};
  int i=1;
   for(;i!=printf("Good\n");){
         printf("\n%d    : ",a[i++]);
         }
         return 0;
}
```

Q.8 What will be the output of given code?

```
#include<iostream>
using namespace std;
int main (){
  int a[5]={10,20,30,40,50};
  int i=4;
  for(;i!=-1;){
         printf("%d\n",a[i--]);
  }
  return 0;
}
```

Q.9 what will be the output of given code?

```
#include<iostream>
using namespace std;
int main(){
   int a[5]={10, 20,30,40,50};
  int i=4;
  for(;i!=-1;){
    printf("%d\n",a[i--]+40);
  }return 0;
}
```

Q.10 what will be the output of given code?

```
#include<iostream>
using namespace std;
int main(){
        int i,j,k;
        i=j=k=1;
        int a[5]={10, 20,30,40,50};
        int b=a [1];
        b+=10;
        a[i+1]=--b+1;
        int c=a[j++];
        int d=a[j+1];
        printf("B= %d\tC=%d\tD=%d\n",b,c,d);
        return 0;
}
```

Q. 11 what will be the output of given code?

```
#include<iostream>
using namespace std;
int main(){
  int a[5]={10,20,30,40,50};
   int *ptr;
   ptr=a;
   for(int i=0; i<5;i++){
        *ptr=*ptr+10;
        printf("%d\n",a[i]);
        ptr++;
        }
        return 0;
}
```

Q.12 what will be the output of given code?

```
#include<iostream>
```

```
using namespace std;
int main(){
  int a[5]={10,20,30,40,50};
    int *ptr;
    ptr=a;
    ptr++;
    *ptr=500;
    for(int i=0;i<5;i++){
              printf("%d\n",a[i]);
            }
            return 0;
}
```

Q.14 what will be the output of given code?

```
#include<iostream>
using namespace std;
int main(){
   int a[5]={10,20,30,40,50};
   int *ptr,*ptr1;
   ptr=a;
   ptr1=ptr;
   *ptr1=100;
   ptr=ptr+2;
   *ptr=600;
   ptr1=ptr1+2;
   *ptr1=700;
   printf("display the array values\n");
   for(int i=0; i<5; i++){
            printf("%d\n",a[i]);
   }
    return 0;
}
```

Q.15 what will be the output of given code?

```
#include<iostream>
using namespace std;
int main(){
 int a[5]={10,20,30,40,50};
  for(int i=0; i<5; i++){
        printf("%d\t%u\n",*(a+i),(i+a));
   }
   return 0;
}
```

Q 16. what will be the output of given code?

```
#include<stdio.h>
int main(){
        int a[]={10,20,30,40,50};
        int *ptr;
        ptr=a;
        ptr=ptr+2;
        int *ptr1;
        ptr1=ptr;
        printf("%d\n",*ptr1);
        return 0;
}
```

Q 17 .what will be the output of given code?

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

Q.18 what will be the output of given code?

```
#include<stdio.h>
int main()
{
   static int sub[5] = {10,20,30,40,50};
   int i;
  for (i = 0; i <= 4; i++);
  {
     if(i<=4)     {
       sub[i] = i * i;
       printf ("%d\n", sub[i]);
     }
  }

           return 0;
}
```

Q.19 what will be the output of given code?

```
#include<stdio.h>
int main()
{
   static int sub[5] = {10,20,30,40,50};
   int i;
  for (i = 0; i <= 4; i++);
  {
     if(i<=4)
     {
       sub[i] = i * i;
       printf ("%d\n", sub[i]);
     }
  }
   return 0;
}
```

Q.20 what will be the output of given code?

```
#include<stdio.h>
int main(){  static int a[5];
   int i;
   for(i=0;i<=4;i++){
         printf("%d\t%d\n",a[i],i);
         }
        return 0;
}
```

Q.21 what will be the output of given code?

```
#include<stdio.h>
int main()
{    int i, j = 10, arrsize;
     int arr[arrsize];
     if (j == 10)
       arrsize = 20 ;
       else
        arrsize = 40 ;
        for (i = 0 ; i < arrsize; i++)
         arr[i] = 100;
       printf("Display the array content\n");
        for(i=0; i<arrsize; i++){ printf("%d\n",arr[i]);
                     }
          return 0;
}
```

Q. 22 what will be the output of given code?

```
#include<stdio.h>
int main()
{
   int arr1[10],arr2[10], i;
   for (i = 0 ; i <= 9 ; i++)
   {  arr1[i]= 'A' + i;
      arr2[i] = 'a' + i;
      printf ("%d\n", arr2[i]-arr1[i]);
   }
        return 0;
}
```

Q. 23 what will be the output of given code?

```
#include<stdio.h>
int main()
{
    static int b[] = {10,20,30,40,50};
    int i;
    for (i = 0 ; i <= 4 ; i++)
     printf ("%d\t", i[b]);
        return 0;
}
```

Q. 24 what will be the output of given code?

```
#include<stdio.h>
int main()
{   static int array[10] = {1,2,3,4,5,6};
    int i;
     for (i = 0 ; i <= 9 ; i++)
        printf ("%d\t",array[i]);
          return 0;
}
```

Q. 25 what will be the output of given code?

```
#include<stdio.h>
int main()
{  static int a[]={2,3,4,5,6};
   int i;
```

```
  for (i = 5 ; i > 0;)
    printf ("%d ",a[--i]);
       return 0;
}
```

Q. 26 what will be the output of given code?

```
#include<stdio.h>
int main()
{ static int a[5] = {5,10,15,20,25};
  int i, j, m, n;
  i = ++a[1];
  j = a[1]++;
  printf ("i = %d j = %d a[1] = %d\n", i, j, a[1]) ;
  i = 1 ;
  m = a[i++];
  printf ("i = %d m = %d\n", i, m) ;
  i = 2 ;
  n = a[++i];
  printf ("i = %dn = %d", i,n);
   return 0;
}
```

Q. 27 what will be the output of given code?

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

Q. 28 what will be the output of given code?

```
#include<stdio.h>
int main()
{
  static float a[] = {13.24,1.5,1.5, 5.4,3.5};
  float *j, *k;
  j = a ;
  k=a+4 ;
  j=j*2;
  k=k/2 ;
  printf ("%f%f",*j,*k);
        return 0;
}
```

Q. 29 what will be the output of given code?

```
#include<stdio.h>
int main()
{
  int n[25];
   n[0] = 100;
   n[24] = 200;
   printf ("%d %d", *n, *(n + 24) + *( n + 0)) ;
   return 0;
}
```

Q. 30 what will be the output of given code?

```
#include<stdio.h>
int main()
{ static int b[] = {10,20,30,40,50};
  int i, *k;
  k=&b[4]-4 ;
  for (i = 0 ; i <= 4 ; i++)
  {   printf ("%d\t",*k);
       k++;
  }
          return 0;
}
```

Q 31 . what will be the output of given code?

```
#include<stdio.h>
int main()
{ static int a[] = {2,4,6,8,10};
   for(int i=0;i<=4; i++){
        *(a+i)=a[i ] + i[a];
    printf ("%d\t", *(i + a)) ;
  }
  return 0;
}
```

Q. 31 what will be the output of given code

```
#include<stdio.h>
int main()
{
 int arr[] = { 0,1,2,3,4};
  int i, *p;
  for (p = arr, i = 0 ; p + i <= arr + 4 ; p++, i++)
     printf ("%d\t",*(p + i)) ;
          return 0;
}
```

Q 32. what will be the output of given code?

```
#include<stdio.h>
int main()
{   int arr[] = {0,1,2,3,4} ;
   int i, *ptr;
    for (ptr = arr + 4 ; ptr = arr; ptr--)
```

```
            printf ("%d ",*ptr);
          return 0;
}
```

Q. 33 what will be the output of given code?

```
#include<stdio.h>
int main()
{    int arr[] = {0,1,2,3,4} ;
     int *ptr, i;
     for (ptr = arr + 4 ; ptr >= arr; ptr--)
         printf ("%d\t", arr [ptr-arr]);
           return 0;
}
```

Q.34 what will be the output of given code?

```
#include<stdio.h>
int main()
{
   static int a[] = {0,1,2,3,4} ;
     static int *p[] = {a, a +1 , a + 2, a + 3, a + 4} ;
     int **ptr = p;
     printf ("%d\t%d\n", a, *a) ;
     printf("%d\t%d\t%d\n", p,*p,**p);
     printf("%d\t%d\t%d\n", ptr,*ptr, **ptr);
           return 0;
}
```

Q. 35 what will be the output of given code?

```
#include<stdio.h>
int main()
{ static int a[]= { 0,1,2,3,4};
    static int *p[ ] = {a, a + 1, a + 2, a + 3, a + 4} ;
    int **ptr = p;
    ptr++;
```

```
   printf ("%d %d %d\n", ptr - p, *ptr - a, **ptr);
    *ptr++;
     printf ("%d %d %d\n", ptr - p, *ptr - a, **ptr);
     *ptr++;
     printf("%d %d %d\n", ptr - p, *ptr - a, **ptr) ;
     *++ptr;
     printf ("%d %d %d\n", ptr - p, *ptr - a, **ptr);
     ++*ptr;
     printf ("%d %d %d\n", ptr - p, *ptr - a, **ptr);
          return 0;
}
```

Q 36. what will be the output of given code?

```
#include<stdio.h>
int main()
{  static int a[] = {0, 1,2,3,4};
   static int*p[] = {a, a+1 , a + 2, a + 3, a + 4} ;
    int **ptr;
    ptr = p;
    **ptr++;
    printf ("%d %d %d\n", ptr - p, *ptr - a, **ptr);
    *++*ptr;
    printf ("%d %d %d\n", ptr - p, *ptr - a, **ptr);
    ++**ptr;
    printf ("%d %d %d\n", ptr - p, *ptr - a, **ptr);
         return 0;
}
```

Q. 37 what will be the output of given code?

```
#include<stdio.h>
int main()
{       static int n[3][3] = {   2,4,3 ,
                                 6, 8, 5,
                              3, 5,1
                    };
printf ("%d\t%d\t%d", n, n[2], n[2][2]);
        return 0;
}
```

Q. 38 what will be the output of given code?

```
#include<stdio.h>
int main()
{
   static int n[3][3] = {2,4,3,6, 8, 5,3,5,1};
    int i, j ;
    for(i = 2; i = 0;i--) {
      for(j = 2; j = 0;j--)
           printf ("%d %d\n", n[i][j], *(*( n + i) + j)) ;
      }
        return 0;
}
```

Q 39 . what will be the output of given code?

```
#include<stdio.h>
int main()
{
   static int a[3][3] = { 1,2,3, 4, 5,6, 7, 8,9 };
    int *ptr[3] = {a[0],a[1],a[2]};
    int **ptr1 = ptr;
    int i;
   for (i = 0 ; i <= 2 ; i++)
      printf ("%d\t",*ptr[i]);
      printf ("\n") ;
    for (i = 0; i <= 2; i++)
      printf ("%d\t",*a[i]);
      printf ("\n");
    for (i = 0; i <= 2; i++)
    {
      printf ("%d\t",**ptr1);
      ptr1++;
    }
         return 0;
}
```

Q 40 . what will be the output of given code?

```
#include<stdio.h>
int main()
{
   static int t[3][2][4] = {
                              {
                          2,4,3,6,
                          1,6,7,9
                        },
                        {
                          8,2,1,1,
                         2, 3, 7, 3
                       },
                      {
                        1,6, 2,4,
                         0, 7, 9,5
                      }
                   } ;
                 printf("%d\t%d", t[2][1][3],*( *( *( t + 2) + 1 ) + 3)) ;
              return 0;
}
```