Calling Functions with Arrays

- In C programming, a single array element or an entire array can be passed to a function.
- Also, both one-dimensional and multi-dimensional array can be passed to function as argument.

Passing a single element of an array to function

```
#include <stdio.h>
void display(int a)
{ printf("%d",a);}


int main()
{int c[]={2,3,4};
display(c[2]);            //Passing array element c[2] only.
return 0;}
```

Output :   4

Passing Entire One-Dimensional Array to a Function

- While passing arrays to the argument, the name of the array is passed as an argument (i.e., starting address of memory area is passed as argument).

Passing Entire One-Dimensional Array to a Function

- Write a C program to pass an array containing age of 6 persons to a function. This function should find average age and display the average age in main function.

Passing Entire One-Dimensional Array to a Function

```
#include <stdio.h>
float average(float a[]);
int main()
{
float avg, c[]={23.4, 55, 22.6, 3, 40.5, 18};
avg=average(c);/* Only name of array is passed as argument. */
printf("Average age=%.2f",avg);
return 0;
}
float average(float a[])
{int i;float avg, sum=0.0;
for(i=0;i<6;i++)
    {sum+=a[i];}
avg =(sum/6);
return avg;
}
```

Passing Multi-dimensional Arrays to Function

- To pass two-dimensional array to a function as an argument, starting address of memory area reserved is passed as in one dimensional array.

Passing Multi-dimensional Arrays to Function

```
#include <stdio.h>
void Function(int c[2][2]);
int main()
{int c[2][2],i,j;
 printf("Enter 4 numbers:\n");
 for(i=0;i<2;++i)for(j=0;j<2;++j)
    {scanf("%d",&c[i][j]);}
   Function(c);     /*passing multi-dimensional array to function */
   return 0;
  }
void Function(int c[2][2])
{/* Instead to above line, void Function(int c[][2]){ is also valid */
int i,j;
printf("Displaying:\n");
for(i=0;i<2;++i)
  for(j=0;j<2;++j)
      printf("%d\n",c[i][j]);
}
```