

C-96 → Passing an Array as an argument to function in C

Array → When we want to pass list of values to a function we use array.

Syntax:-

return type		datatype
int	function_name	(int [])

function declaration	←	return type	function_name	(datatype [])
function call	←		function_name	(array_name)
		return type	function_name	(datatype arrayname)

Eg:- → int avg (int []); ⇒ function declaration

Example Program:

```
int avg (int [], int);
```

```
void main()
```

```
{  
    int average = 0;
```

```
    int marks[5] = {10, 15, 20, 30, 45};
```

```
    average = avg(marks, 5);
```

→ array size

```
    printf("Average is %.d", average);
```

```
    printf("%.d", sizeof(marks));
```

→ 20 bytes

base address is passed & it acts like pointer inside memory
eg: call by reference

```
int avg (int marks[], int a);
```

acts like pointer

call by reference

call by value

```
{  
    int sum = 0; average = 0; i;
```

```
    for (i = 0; i < a; i++)
```

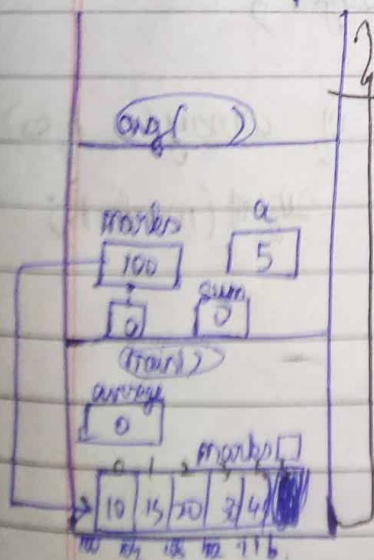
```
    {  
        sum = sum + marks[i];
```

```
    }  
    average = sum / a;
```

```
    return average;
```

```
    printf("%.d", sizeof(marks));
```

→ 4 bytes



Program

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
int avg(int[], int);
```

```
void main()
```

```
{
```

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

```
int average = 0;
```

```
int size;
```

```
size = sizeof(marks) / sizeof(marks[0]);
```

```
average = avg(marks, size);
```

```
printf("Average = %.d", average);
```

```
printf("Inside main size of  
array is (in bytes): %.d", sizeof(marks));
```

```
int avg(int marks[], int size)
```

```
{
```

```
int i, sum = 0, average = 0;
```

```
for(i = 0; i < size; i++)
```

```
{  
sum = sum + marks[i];
```

```
}
```


```
average = sum / size;
```

```
return average;
```

```
printf("Inside avg function size of array (in bytes) is %d",  
sizeof(marks));
```

PROBLEM 1:

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  /** 1 - PASSING ARRAY AS AN ARGUMENT TO FUNCTION **/
4  int marks(int[],int);
5  int main()
6  {
7      int marks[]={10,20,30,40,50};
8      int size=0,average=0;
9      size=sizeof(marks)/sizeof(marks[0]);
10     average=avg(marks,size);
11     printf("Average is %d\n",average);
12     getch();
13 }
14 int avg(int marks1[],int size1)
15 {
16     int i,sum=0,average=0;
17     for(i=0;i<size1;i++)
18     {
19         sum=sum+marks1[i];
20     }
21     average=sum/size1;
22     return average;
23 }
```

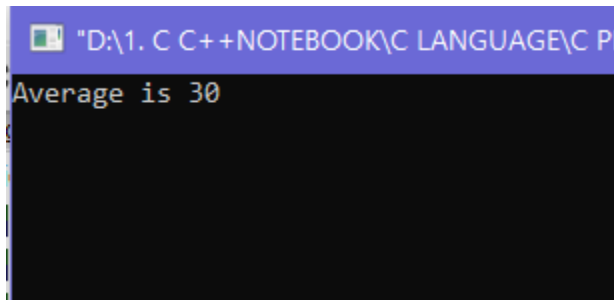
 "D:\1. C NOTEBOOK\C LANGUAGE\C PROGRAMS\PART 5_Jennys Lecture

Average is 30

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  /** 1 - PASSING ARRAY AS AN ARGUMENT TO FUNCTION **/
4  int marks(int*,int); //or int marks(int[],int)
5  int main()
6  {
7      int marks[]={10,20,30,40,50};
8      int size=0,average=0;
9      size=sizeof(marks)/sizeof(marks[0]);
10     average=avg(marks,size); //we are passing only base address of marks
11     printf("Average is %d\n",average);
12     getch();
13 }
14 int avg(int* marks1,int size1) //or int avg(int marks1[],int size1)
15 {
16     int i,sum=0,average=0;
17     for(i=0;i<size1;i++)
18     {
19         sum=sum+marks1[i];
20     }
21     average=sum/size1;
22     return average;
23 }
24
25

```



The screenshot shows a Windows-style window titled "D:\1. C C+++NOTEBOOK\C LANGUAGE\C P". The window contains a black console area with the text "Average is 30" displayed in white. The text is centered horizontally and appears to be the output of the C++ program shown in the code block above.