



## Pass Array to Functions in C

Last Updated : 04 Dec, 2023

In C, the whole array cannot be passed as an argument to a function. However, you can pass a pointer to an array without an index by specifying the array's name.

*Arrays in C are always passed to the function as pointers pointing to the first element of the array.*

### Syntax

In C, we have three ways to pass an array as a parameter to the function. In the function definition, use the following syntax:

```
return_type foo ( array_type array_name[size], ... );
```

Mentioning the size of the array is optional. So the syntax can be written as:

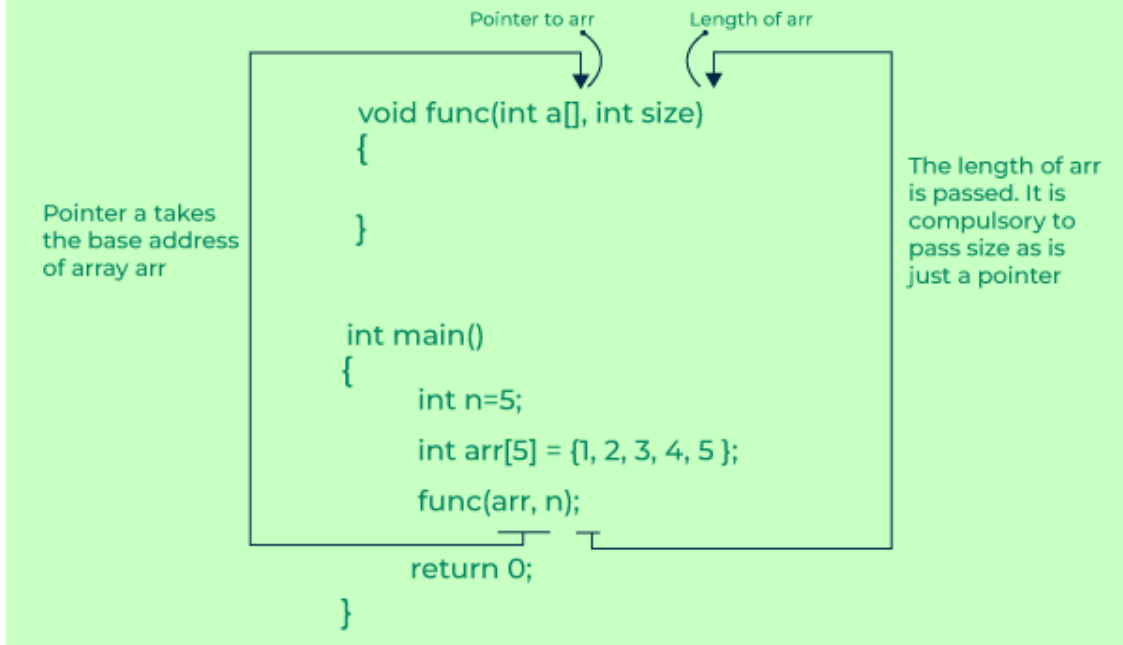
```
return_type foo ( array_type array_name[], ... );
```

In both of the above syntax, even though we are defining the argument as array it will still be passed as a pointer. So we can also write the syntax as:

```
return_type foo ( array_type* array_name, ... );
```

But passing an array to function results in [array decay](#) due to which the array loses information about its size. It means that the size of the array or the number of elements of the array cannot be determined anymore.

# Passing Array to Function in C



## Examples of Passing Array to Function in C

### Example 1: Checking Size After Passing Array as Parameter

C

```
// C program to pass the array as a function and check its size
#include <stdio.h>
#include <stdlib.h>

// Note that arr[] for fun is just a pointer even if square
// brackets are used. It is same as
// void fun(int *arr) or void fun(int arr[size])
void func(int arr[8])
{
    printf("Size of arr[] in func(): %d bytes",
           sizeof(arr));
}

// Drive code
int main()
{
    int arr[8] = { 1, 2, 3, 4, 5, 6, 7, 8 };

    printf("Size of arr[] in main(): %dbytes\n",
```

```

        sizeof(arr));

    func(arr);

    return 0;
}

```

## Output

```

Size of arr[] in main(): 32bytes
Size of arr[] in func(): 8 bytes

```

As we can see,

- The size of the arr[] in the main() function (where arr[] is declared) is 32 bytes which is  $= \text{sizeof(int)} * 8 = 4 * 8 = 32$  bytes.
- But in the function where the arr[] is passed as a parameter, the size of arr[] is shown as 8 bytes (which is the size of a pointer in C).

It is due to the fact that the array decays into a pointer after being passed as a parameter. One way to deal with this problem is to pass the size of the array as another parameter to the function. This is demonstrated in the below example.

## Example 2: Printing Array from the Function

---

### C

```

// C program to illustrate the use of sizeof operator in C
#include <stdio.h>

// function to print array
void printArray(int arr[], int size)
{
    printf("Array Elements: ");
    for (int i = 0; i < size; i++) {
        printf("%d ", arr[i]);
    }
}

// driver code
int main()
{

```

```

    int arr[8] = { 12, 4, 5, 3, 7, 8, 11, 45 };

    // getting the size of array
    int size = sizeof(arr) / sizeof(arr[0]);

    printArray(arr, size);

    return 0;
}

```

## Output

```

Array Elements: 12 4 5 3 7 8 11 45

```

This size parameter may not be needed only in the case of '\0' terminated character arrays as size can be determined by checking the end of the string character. The below example demonstrates the following.

## Example 3: Printing Array of Characters (String) using Function

---

### C

```

// C Program to print the string using a function
#include <stdio.h>

// function to print the string
void printString(char* str)
{
    printf("Array of Characters: ");

    int i = 0;
    while (str[i] != '\0') {
        printf("%c ", str[i]);
        i++;
    }
}

// Driver program
int main()
{
    char arr[] = "String";

    printString(arr);
}

```

```
    return 0;
}
```

## Output

```
Array of Characters: S t r i n g
```

As we can see, we were able to deduce the end of the string (character array) using a NULL Character. Moreover, the %s specifier in printf() function uses the same principle so we can directly use this to print string.

## Predict the Output of the Below C Programs

### Program 1:

---

#### C

```
#include <stdio.h>

void fun(int arr[], unsigned int n)
{
    int i;
    for (i = 0; i < n; i++)
        printf("%d ", arr[i]);
}

// Driver program
int main()
{
    int arr[] = { 1, 2, 3, 4, 5, 6, 7, 8 };
    unsigned int n = sizeof(arr) / sizeof(arr[0]);
    fun(arr, n);
    return 0;
}
```

### Program 2:

---

## C

```
#include <stdio.h>
void fun(int* arr)
{
    int i;
    unsigned int n = sizeof(arr) / sizeof(arr[0]);
    for (i = 0; i < n; i++)
        printf("%d ", arr[i]);
}

// Driver program
int main()
{
    int arr[] = { 1, 2, 3, 4, 5, 6, 7, 8 };
    fun(arr);
    return 0;
}
```

*Hint: For the 64-bit processor, the size of the pointer is 8 bytes and the size of the integer is 4 bytes.*

### Program 3:

---

## C

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

void fun(char* arr)
{
    int i;
    unsigned int n = strlen(arr);
    printf("n = %d\n", n);
    for (i = 0; i < n; i++)
        printf("%c ", arr[i]);
}

// Driver program
int main()
```

```

{
    char arr[] = "geeksquiz";
    fun(arr);
    return 0;
}

```

## Program 4:

---

### C

```

#include <stdio.h>
#include <string.h>

void fun(char* arr)
{
    int i;
    unsigned int n = strlen(arr);
    printf("n = %d\n", n);
    for (i = 0; i < n; i++)
        printf("%c ", arr[i]);
}

// Driver program
int main()
{
    char arr[]
        = { 'g', 'e', 'e', 'k', 's', 'q', 'u', 'i', 'z' };
    fun(arr);
    return 0;
}

```

**NOTE:** The character array in the above program is not '\0' terminated.  
(See [this](#) for details)

## Frequently Asked Questions (FAQs)

**Q1.** Why can't we use NULL to terminate the integer or float array?

Answer:

*In C, the null by default is equivalent to 0. So if we want to use NULL as an ending point for an integer or float array, we won't be able to use 0 as a value of the array element. In some specific cases, it may work but in general, it is meaningless to sacrifice one of the most commonly used digit for this purpose.*

**Q2. Why does C not allow passing an array by value to the function?**

Answer:

*The array names in C are inherently treated as a pointer to the first element of the array in most contexts including passing as a parameter. So, when we use the array name to pass it as a parameter, the value of the array name is actually copied to the formal parameter of the function (as happens in call by value), but the value that is copied is the address of the first element. So it seems like the arrays are passed as reference.*

**Related Articles:**

- [Pass a 2D Array as a Parameter in C](#)
- [Do Not Use sizeof for Array Parameters](#)
- [Difference Between Pointer and Array in C?](#)

Summer-time is here and so is the time to skill-up! More than 5,000 learners have now completed their journey from **basics of DSA to advanced level development programs** such as Full-Stack, Backend Development, Data Science.

And why go anywhere else when our [DSA to Development: Coding Guide](#) will help you master all this in a few months! Apply now to our [DSA to](#)



[Development Program](#) and our counsellors will connect with you for further guidance & support.

[Suggest improvement](#)

75

## Previous

Jagged Array or Array of Arrays in C with Examples

## Next

How to pass a 2D array as a parameter in C?

[Share your thoughts in the comments](#)

Add Your  
Comment

## Similar Reads

Pass Array to Functions in C++

C++ Functions - Pass By Reference

How to pass a 2D array as a parameter in C?

How to pass an array by value in C ?

How to Pass an Array Pointer as a Function Argument in C++?



GeeksforGeeks

Article Tags :

[CBSE - Class 11](#)

[c++-array](#)

[C++-Functions](#)

[c++-pointer](#)

[school-programming](#)

[C Language](#)

[C++](#)

Practice Tags :

[C++](#)



A-143, 9th Floor, Sovereign Corporate  
Tower, Sector-136, Noida, Uttar Pradesh -  
201305



## Company

About Us  
Legal  
Careers  
In Media  
Contact Us  
Advertise with us  
GFG Corporate Solution  
Placement Training Program

## Languages

Python  
Java  
C++  
PHP  
GoLang  
SQL  
R Language  
Android Tutorial  
Tutorials Archive

## Explore

Hack-A-Thons  
GfG Weekly Contest  
DSA in JAVA/C++  
Master System Design  
Master CP  
GeeksforGeeks Videos  
Geeks Community

## DSA

Data Structures  
Algorithms  
DSA for Beginners  
Basic DSA Problems  
DSA Roadmap  
Top 100 DSA Interview Problems  
DSA Roadmap by Sandeep Jain  
All Cheat Sheets

## Data Science & ML

Data Science With Python  
Data Science For Beginner  
Machine Learning Tutorial  
ML Maths  
Data Visualisation Tutorial  
Pandas Tutorial  
NumPy Tutorial  
NLP Tutorial  
Deep Learning Tutorial

## Python Tutorial

Python Programming Examples  
Python Projects  
Python Tkinter  
Web Scraping  
OpenCV Tutorial  
Python Interview Question  
Django

## DevOps

Git  
AWS  
Docker  
Kubernetes  
Azure  
GCP  
DevOps Roadmap

## System Design

High Level Design  
Low Level Design  
UML Diagrams  
Interview Guide  
Design Patterns  
OOAD  
System Design Bootcamp  
Interview Questions

## Preparation Corner

Company-Wise Recruitment Process  
Resume Templates  
Aptitude Preparation  
Puzzles  
Company-Wise Preparation

## HTML & CSS

HTML  
CSS  
Web Templates  
CSS Frameworks  
Bootstrap  
Tailwind CSS  
SASS  
LESS  
Web Design

## Computer Science

Operating Systems  
Computer Network  
Database Management System  
Software Engineering  
Digital Logic Design  
Engineering Maths

## Competitive Programming

Top DS or Algo for CP  
Top 50 Tree  
Top 50 Graph  
Top 50 Array  
Top 50 String  
Top 50 DP  
Top 15 Websites for CP

## JavaScript

JavaScript Examples  
TypeScript  
ReactJS  
NextJS  
AngularJS  
NodeJS  
Lodash  
Web Browser

## School Subjects

Mathematics  
Physics  
Chemistry  
Biology  
Social Science  
English Grammar  
World GK

## Management & Finance

Management  
HR Management  
Finance  
Organisational Behaviour  
Marketing

## More Tutorials

Software Development  
Software Testing  
Product Management  
SEO - Search Engine Optimization  
Linux  
Excel  
All Cheatsheets

## Free Online Tools

Typing Test  
Image Editor  
Code Formatters  
Code Converters  
Currency Converter  
Random Number Generator  
Random Password Generator

## GeeksforGeeks Videos

DSA  
Python  
Java  
C++  
Web Development  
Data Science  
CS Subjects

@GeeksforGeeks, Sanchhaya Education Private Limited, All rights reserved