How to print size of array parameter in C++?

Difficulty Level : Medium • Last Updated : 10 Mar, 2023

How to compute the size of an array CPP?

C++

```
// A C++ program to show that it is wrong to
// compute size of an array parameter in a function
#include <iostream>
using namespace std;

void findSize(int arr[])
{
  cout << sizeof(arr) << endl;
}

int main()
{
    int a[10];
    cout << sizeof(a) << " ";
    findSize(a);
    return 0;
}
```

Output

```
 40 8
```

Time Complexity: O(1)
Auxiliary Space: O(n) where n is the size of the array.

The above output is for a machine where the size of an integer is 4 bytes and the size of a pointer is 8 bytes.

The **cout** statement inside main prints 40, and **cout** in findSize prints 8. The reason is, arrays are always passed pointers in functions, i.e., findSize(int arr[]) and findSize(int *arr) mean exactly same thing. Therefore the cout statement inside findSize() prints the size of a pointer. See this and this for details.

How to find the size of an array in function?

We can pass a 'reference to the array'.

CPP

```cpp
// A C++ program to show that we can use reference to
// find size of array
#include <iostream>
using namespace std;

void findSize(int (&arr)[10])
{
    cout << sizeof(arr) << endl;
}

int main()
{
    int a[10];
    cout << sizeof(a) << " ";
    findSize(a);
    return 0;
}
```

Output

```
40 40
```

Time Complexity: O(1)

Space Complexity: O(n) where n is the size of array.

The above program doesn't look good as we have a hardcoded size of the array parameter.

We can do it better using templates in C++**.**

CPP

```cpp
// A C++ program to show that we use template and
// reference to find size of integer array parameter
#include <iostream>
using namespace std;

template <size_t n>
void findSize(int (&arr)[n])
{
    cout << sizeof(int) * n << endl;
}

int main()
{
    int a[10];
    cout << sizeof(a) << " ";
    findSize(a);
    return 0;
}
```

Output

```
 40 40
```

Time Complexity: O(1)

Space Complexity: O(n) where n is the size of array.

We can make a generic function as well:

CPP

```cpp
// A C++ program to show that we use template and
// reference to find size of any type array parameter
#include <iostream>
using namespace std;

template <typename T, size_t n>
void findSize(T (&arr)[n])
{
    cout << sizeof(T) * n << endl;
}
```

```
int main()
{
    int a[10];
    cout << sizeof(a) << " ";
    findSize(a);

    float f[20];
    cout << sizeof(f) << " ";
    findSize(f);
    return 0;
}
```

Output

```
40 40
80 80
```

Time Complexity: O(1)

Space Complexity: O(n)

Now the next step is to print the size of a dynamically allocated array.

It's your task man! I'm giving you a hint.

CPP

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

int main()
{
    int *arr = (int*)malloc(sizeof(int) * 20);
    return 0;
}
```

This article is contributed by **Swarupananda Dhua** Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above

145

Related Articles