

Re: How to know the size of array

Source: http://coding.derkeiler.com/Archive/C_CPP/comp.lang.c/2006-03/msg03976.html

- *From:* "John Bode" <john_bode@xxxxxxxxxxxx>
 - *Date:* 24 Mar 2006 06:57:06 -0800
-

manochavishal@xxxxxxxx wrote:

```
void foo(int * array)
```

```
void foo(int *array, size_t len)
```

```
{
```

Now the array size is in `len`.

Thanx for the quick reply.

I think i read somewhere that i can get the size by:

```
(sizeof(array) ) / (sizeof(element))
```

in this case it will be

```
(sizeof(array)) / (sizeof(int))
```

Given the following:

```
int foo[10];
```

sizeof foo will return the total number of bytes taken up by the whole

Re: How to know the size of array

array, and `sizeof foo[0]` will return the number of bytes used by a single array element, so yes, the expression

```
sizeof foo / sizeof foo[0]
```

will give you the number of elements (10) in the array.

Unfortunately, this won't help you if you pass the array to a function. In most contexts*, the type of the array identifier "decays" into a pointer to the base type, and its value is set to the address of the first element in the array. IOW, when you write

```
bar(foo);
```

what actually gets passed to `bar` is a pointer type object, not an array type object:

```
void bar(int *arr)
{
...
}
```

and `sizeof arr` will return the number of bytes used by the pointer, *not* the array it points to, so the `sizeof arr / sizeof arr[0]` trick won't work.

* The only times the array identifier doesn't decay into a pointer is when it's an operand of the `sizeof` and unary `&` operators.

I am not sure about this. Can we get the `sizeof` array to return the size of all its elements.

Can `sizeof` find where the array is getting finished or it will just give me the size of element.

Is there no way to get the size of array without passing the its length.

Not really, no.

Cheers
Vishal

.