What is the type of an int array? [duplicate]

Asked 7 years, 4 months ago Modified 7 years, 4 months ago Viewed 1k times

1

This question already has answers here:
How do I use arrays in C++? (5 answers)
Closed 7 years ago.

I am reading about arrays in C++. I tried the following code:

```
int main()
{
   int a[10];
   int *p;
   p = &a;
}
```

I got compiler error:

```
pointers.cpp:10:6: error: cannot convert ‘int (*)[10]’ to ‘int*’ in assignment
p = &a;
```

In order to understand the array type to be able to assign to a pointer I tried this code:

```
int main()
{
   int a[10];
   int *r[10];
   r = a;
}
```

Compilation error:

```
: error: incompatible types in assignment of ‘int [10]’ to ‘int* [10]’
  r = a;
```

Then I tried this:

```
int main()
{
   int a[10];
   int *r[10];
   r = &a;
}
```

Compilation error:

```
   error: incompatible types in assignment of ‘int (*)[10]’ to ‘int* [10]’
   r = &a;
```

What is the type `int (*)[10]` ?

c++ arrays pointers

Share Edit Follow Flag

edited Dec 19, 2014 at 12:31 user1804599

asked Dec 19, 2014 at 12:27 kalyan 23 ● 7

@dandan78 The question is about C++, not about C. Arrays don't work exactly the same in the two languages. – user1804599 Dec 19, 2014 at 12:32

Just don't try with C++, if you don't know what you are doing. Your house could fall. – nbro Dec 19, 2014 at 12:39

5 Answers

Sorted by: Highest score (default)

3

`int (*)[10]` is a pointer to an array of 10 `int`s. It points at the entire array. In your first code, when you do `&a`, you are getting the address of the entire array, which has this type. If you wanted to store that, you'd have to write:

```
int (*r)[10] = &a;
```

In the code that you tried, `int *r[10]` is instead an array of pointers (rather than a pointer to an array).

An `int*` on the other hand, points at a single `int`. You could, for example, take the address of one of your array elements, `&a[0]`, and it would be an `int*`.

```
int *p = &a[0];
```

In fact, the name of an array often undergoes an implicit conversion called array-to-pointer conversion in which it becomes a pointer to the first element in the array. We typically say that the array *decays* to a pointer to its first element.

```
int *p = a; // Here, a is decaying
```

Share Edit Follow Flag

answered Dec 19, 2014 at 12:31 Joseph Mansfield 105k ● 19 ● 232 ● 314

2

> What is the type of an int array?

There's a different type for each size. For example, the type of an array of ten is `int[10]`. There's also an incomplete type `int[]`, representing an array of unknown size.

> What is the type int (*)[10] ?

That's a pointer to an array of size 10.

In your first snippet, `&a` is a pointer to the array, and you can't assign that to a pointer to a single `int`, since `int` and `int[10]` are incompatible types. In the other examples, `r` is an array of pointers, and you can't assign anything to an array.

You could use either of these:

```
int *p = a;      // implicit array-to-pointer conversion
int *p = &a[0];  // explicitly take address of element
```

to assign a pointer to the first element of the array. Or if you want a pointer to the array (perserving the size in its type), that's

```
int (*p)[10] = &a;
```

The extra `()` indicate that this is a pointer to an array, not an array of pointers.

Share Edit Follow Flag

edited Dec 19, 2014 at 12:39

answered Dec 19, 2014 at 12:33 Mike Seymour 242k ● 26 ● 432 ● 630

2

The right way to go about this is to pass the address of the first element:

```
int a[10];
int *p;
p = &a[0];
```

The others can be accessed by incrementing `p`:

```
p++;
```

Share Edit Follow Flag

answered Dec 19, 2014 at 12:31 karlphillip 89.8k ● 35 ● 240 ● 408

1

You simply have to do

```
int a[10];
int *p;
p = a;
```

assign p the adress of array a; i-e address of its first element.

Share Edit Follow Flag

answered Dec 19, 2014 at 12:34 Ali Kazmi 1,410 ● 8 ● 22

0

As the error says, its `int (*)[10]`, so you should use `int (*r)[10]` instead of `int *r[10]`. The extra `()` are the difference between a pointer to an array of 10 ints and an array of 10 pointers to ints.

Share Edit Follow Flag

answered Dec 19, 2014 at 12:33 Zaskar 471 ● 3 ● 10