C++之返回数组指针

因为数组不能被拷贝，所以函数不能返回数组。不过函数可以返回数组的指针或者引用，今天我们一起来学习下C++下的几种返回数组指针的方法

一、类型别名

这是返回数组指针最直接的一种方法了

```
#include<iostream>

using namespace std;

typedef int arrT[10];//类型别名，表示含有10个整形的数组
//using arrT = int[10]; //arrT的等价声明,c++11版本新加的

int arry[10] = {1,3,5,4,5,11,7,13,9,20};
arrT* func()
{
    return &arry;
}

int main()
{
    int (*p)[10] =func();
    for(int i=0;i<10;i++)
    {
        cout<<*(*p + i)++<<" "<<endl;
    }
    return 0;
}
```

打印结果：

```
shun@ubuntu: ~/c++-prime/chapter6
shun@ubuntu:~/c++-prime/chapter6$ g++ return_ArrPoint1.cpp
shun@ubuntu:~/c++-prime/chapter6$ ./a.out
1 2 3 4 5 6 7 8 9 10
shun@ubuntu:~/c++-prime/chapter6$
```

二、后接数组维度

Type (*function(parameter_list))[dimension]

类似于其他数组的声明，Type表示元素的类型，dimension表示数组的大小。

（*function(parameter_list)）两端的括号必须存在，就像我们定义p2时两端必须有括号一样，如果没有这对括号，函数的返回类型将是指针的数组。

int arr[10];　　//arr是一个含有10个整数的数组

int *p1[10];　　//p1是一个含有10个指针的数组

int (*p2)[10] = &arr;　　//p2是一个指针，指向含有10个整数的数组。

源码：

```
#include<iostream>

using namespace std;

int arry[10] = {1,2,3,4,5,6,7,8,9,10};

int (*func())[10]
{
    return &arry;
}

int main()
{
    int (*p)[10] =func();
    for(int i=0;i<10;i++)
    {
        cout<<(**p)++<<" ";
    }
        cout<<endl;
    return 0;
}
```

打印结果：

```
shun@ubuntu: ~/c++-prime/chapter6
shun@ubuntu:~/c++-prime/chapter6$ g++ return_ArrPoint2.cpp
shun@ubuntu:~/c++-prime/chapter6$ ./a.out
1 2 3 4 5 6 7 8 9 10
shun@ubuntu:~/c++-prime/chapter6$
```

三、使用尾置返回类型

在C++11中新增了一种方法，任何函数的定义都能使用尾置返回，但是这种形式对于返回类型教复杂的类型最有效，比如返回类型是数组的指针或者是数组的引用时。尾置返回类型跟在形参列表后面并以一个->符号开头。为了表示函数真正的返回类型跟在形参列表之后，在本应该出现返回类型的地方仿置一个auto。

源代码：

```
#include<iostream>

using namespace std;

int arry[10] = {1,2,3,4,5,6,7,8,9,10};

auto func() -> int(*)[10]
{
    return &arry;
}

int main()
{
    int (*p)[10] =func();
    for(int i=0;i<10;i++)
    {
        cout<<(**p)++<<" ";
    }
        cout<<endl;
    return 0;
}
```

打印结果：

```
shun@ubuntu: ~/c++-prime/chapter6
shun@ubuntu:~/c++-prime/chapter6$ g++ return_ArrPoint3.cpp -std=c++11
shun@ubuntu:~/c++-prime/chapter6$ ./a.out
1 2 3 4 5 6 7 8 9 10
shun@ubuntu:~/c++-prime/chapter6$
```

四、使用decltype

如果我们知道函数返回的指针将指向哪个数组，就可以使用decltype关键字声明返回类型。

源代码：

```
#include<iostream>

using namespace std;

int arry[10] = {1,2,3,4,5,6,7,8,9,10};

decltype(arry) *func()
{
    return &arry;
}

int main()
{
    int (*p)[10] =func();
    for(int i=0;i<10;i++)
    {
        cout<<(**p)++<<" ";
    }
        cout<<endl;
    return 0;
}
```

打印结果：

```
shun@ubuntu: ~/c++-prime/chapter6
shun@ubuntu:~/c++-prime/chapter6$ g++ return_ArrPoint4.cpp -std=c++11
shun@ubuntu:~/c++-prime/chapter6$ ./a.out
1 2 3 4 5 6 7 8 9 10
shun@ubuntu:~/c++-prime/chapter6$
```

decltype并不负责把数组类型转换成对应的指针，所以decltype的结果是一个数组，要想func返回指针还必须在函数声明时加一个*符号。