

10.6 使用指针从函数返回一个数组

前面一节，我们分析了怎样把数组作为参数传递给函数。本节换个方向讨论数据的转换：从函数返回一个数组。

严格地说，无法直接从函数返回一个数组。但是，可以让函数返回一个指向任何数据结构的指针，当然也可以是一个指向数组的指针。记住，声明必须在使用之前。一个声明的例子是：

```
int (*paf())[20];
```

这里，`paf` 是一个函数。它返回一个指向包含 20 个 `int` 元素的数组的指针。它的定义可能如下：

```
int (*paf())[20] {  
  
    int (*pear)[20];    /* 声明一个指向包含 20 个 int 元素的数组的指针 */  
    pear = calloc(20, sizeof(int));  
    if(!pear) longjmp(error, 1);  
    return pear;  
}
```

你用下面这样的方法来调用函数：

```
int (*result)[20];    /* 声明一个指向包含 20 个 int 元素的数组的指针 */  
...  
result = paf();        /* 调用函数 */  
(*result)[3] = 12;    /* 访问结果数组 */
```

或者玩个花样，定义一个结构：

```
struct a_tag {  
    int array[20];  
} x, y;  
struct a_tag my_function() { ... return y }
```

用下面的方法来使用：

```
x = y;  
x = my_function();
```

如果要访问数组中的元素，可以用下面的方法：

```
x.array[i] = 38;
```

千万要注意，不能从函数中返回一个指向函数的局部变量的指针（详见第 2 章）。