## 实验六 Array I

## 学号: 09017423 姓名：杨彬

### 实验目的

Objectivities

1. Using rand to generate random numbers and using srand to seed the random-number generator.

2. Declaring, initializing and referencing arrays.

3. The follow-up questions and activities also will give you practice:

4. Remembering that arrays begin with subscript 0 and recognizing off-by-one errors.

5. Preventing array out-of-bounds errors.

6. Using two-dimensional arrays.

目标

1。使用RAND随机数的生成和使用srand种子随机数发生器。

2。声明、初始化和引用数组。

3.后续的问题和活动也会给你练习：

4。记住数组以下标0开头，并识别一个错误。

5。防止数组越界错误。

6。使用二维数组。

### 实验内容

Exp1

Description of the Problem

Write a program that simulates the rolling of two dice. The program should call rand to roll the first die, and should call rand again to roll the second die. The sum of the two values should

then be calculated. [Note: Each die has an integer value from 1 to 6, so the sum of the two values will vary from 2 to 12, with 7 being the most frequent sum and 2 and 12 being the least frequent sums.] Figure L 7.1 shows the 36 possible combinations of the two dice. Your program should roll the two dice 36,000 times. Use a one-dimensional array to tally the numbers of times each sum appears. Print the results in a tabular format. Also, determine if the totals are reasonable (i.e., there are six ways to roll a 7, so approximately one sixth of all the rolls should be 7).

对问题的描述

编写一个模拟两个骰子滚动的程序。程序应该调用rand来滚动第一个色子，并且应该再次调用rand来滚动第二个色子。两个值的和应该

然后计算。[注：每一个骰子都有一个从1到6的整数值，所以这两个值的总和将从2到12，7是最频繁的和，2和12是最少的总数。图l 7.1显示了这两个骰子的36种可能的组合。你的程序应该把这两个骰子掷36000次。使用一维数组来计算每个和出现的次数。以表格格式打印结果。同时，确定总数是否合理（也就是说，有六种滚动7的方法，所以大约六分之一的卷应该是7）。

Follow-Up Questions and Activities

1. Why is the variable SIZE initialized to 13 when there are only 11 possible die-roll outcomes?

2. What happens if the < operator on line 47 of the program template is changed to <=?

3. What happens if the elements of array sum are not initialized to zero? Try running the program without initializing the array. Show your results.

4. Modify the program to use a two-dimensional array similar to the diagram in Figure L 7.1. Now, rather than counting the number of times each sum appears, increment the correct cell in the array. Print this array with the number of times each dice combination occurred. A sample output may look like the

跟进问题和活动

1。为什么只有11种可能的骰子结果时，变量的大小被初始化为13？

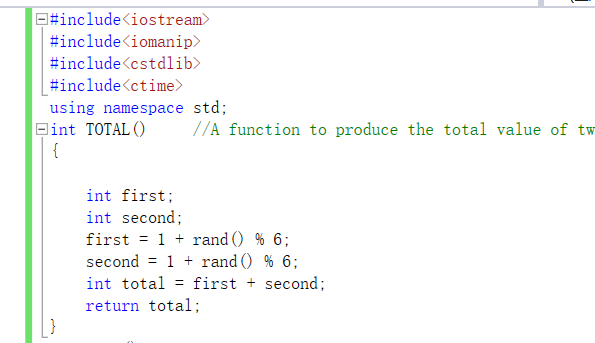
2。如果程序模板的第47行上的<运算符更改为< = >，会发生什么？

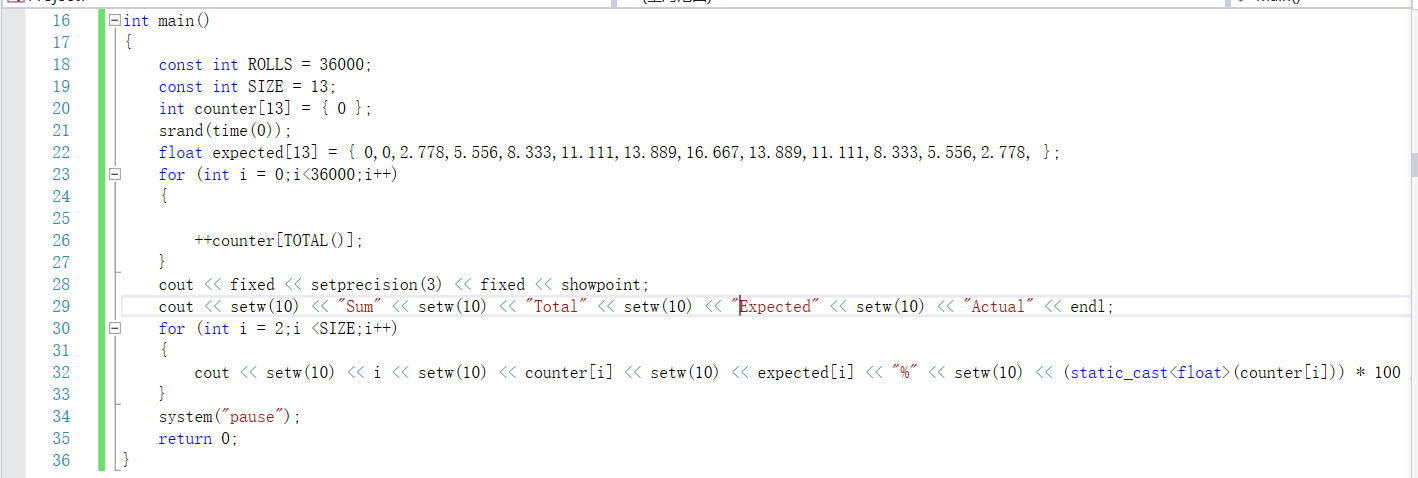
三.如果数组和的元素没有初始化为零会发生什么？尝试在不初始化数组的情况下运行程序。展示你的结果。

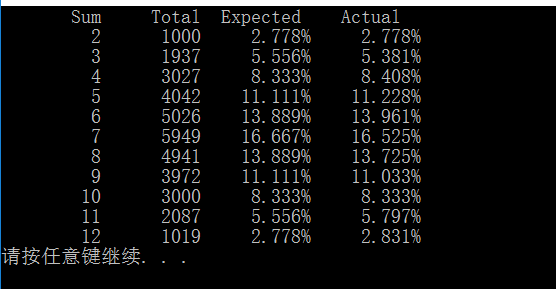
4。修改程序使用二维数组，如图l 7.1所示。现在，而不是计算每个总和出现的次数，在数组中增加正确的单元格。用每个骰子组合的次数打印这个数组。示例输出可能类似于

### 实验代码及结果：

Exp1



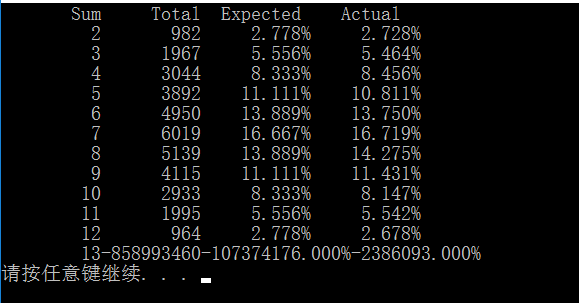


结果

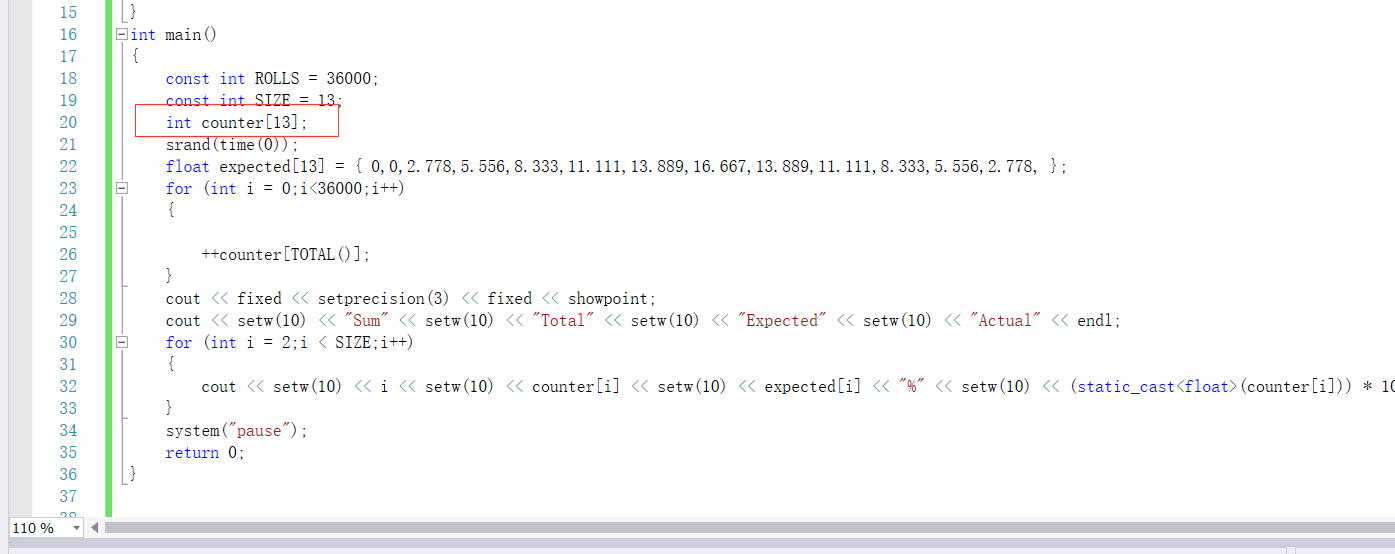
Follow-up1 .为了让程序更加自然

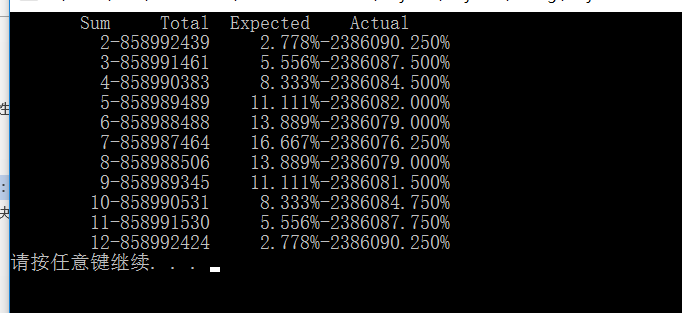
Follow-up2



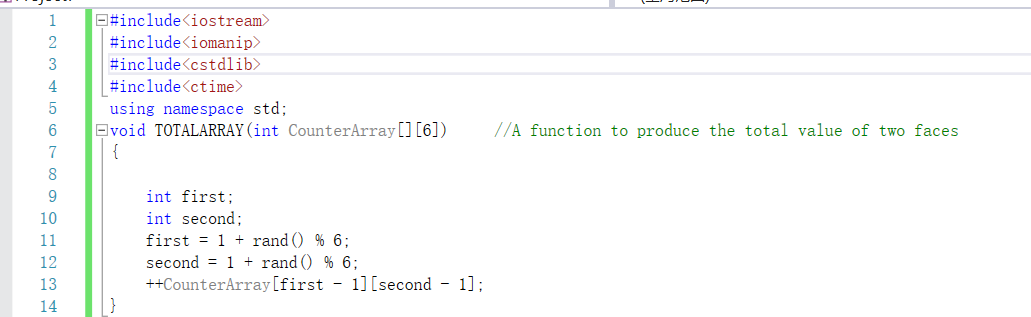


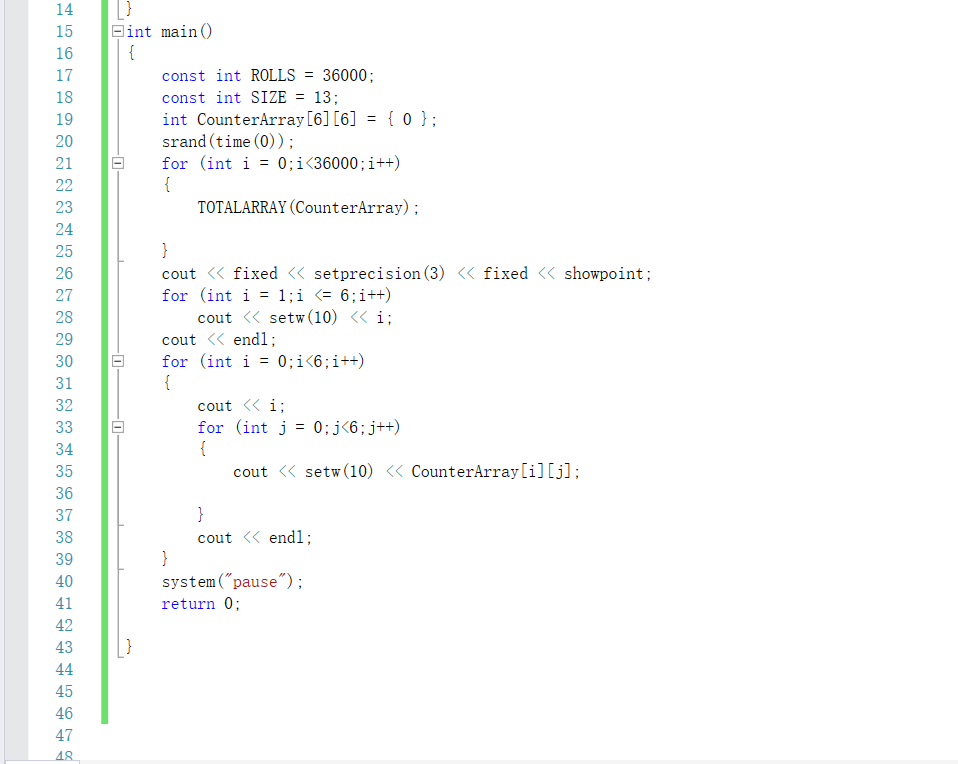
Follow-up3

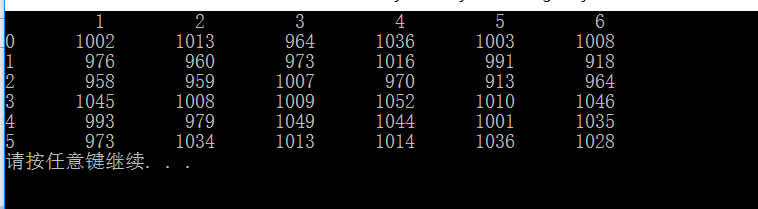




Follow-up4



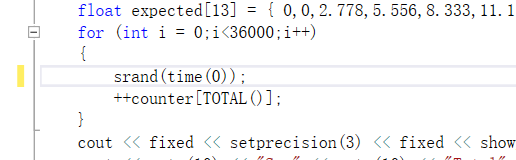


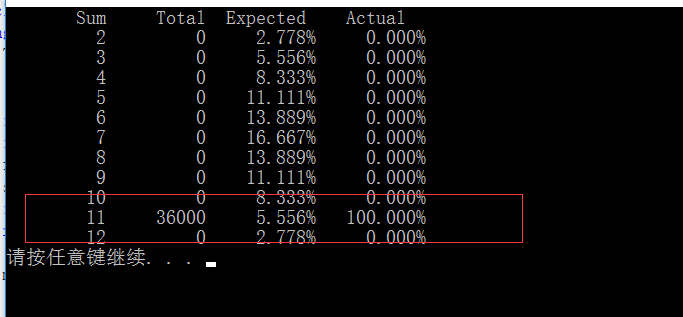


### 遇到的问题及解决过程

Exp1 一开始我将srand(time(0))这个语句放在循环体之中。

结果得到的都是同一个数

如下



我对这个结果感到非常奇怪。

问了知道问题的愿意，我询问助教，查找资料得到了一个不成熟的结论就是srand(time(0))语句根据时间产生一个序列的种子，但是如果放在for循环里的话，rand每一次都从种子序列的首端开始参照，这也就导致了产生的都是同一个数。（这个结论不一定正确）

总之在使用srand()语句的时候我应当把它放在循环体的外部。

Follow-up 123:nothing

Follow-up4这里主要涉及到一个在二维数组中的参数传递的问题，我一开始传递的参数是

结果导致编译错误，

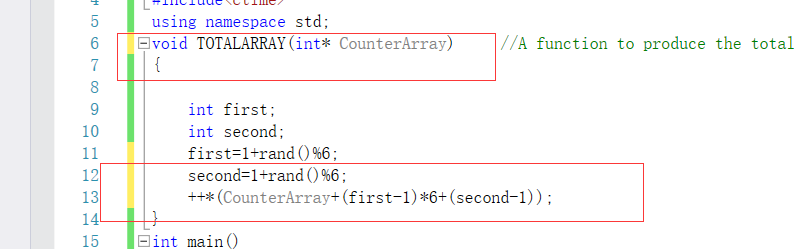
查阅课本后，知道了传递二维数组给函数的时候，不能忽略第二个以及第二个以后的大小 ，

因此修改成问题就得到解决。

同时我才用另外一种传递参数的方法



然后对函数中的代码进行改动也能够得到结果



### 实验体会

Exp1

在这个实验中，我掌握了用rand（）函数来产生随机数的方法，具体表示为

a + rand()%b 其中a表示区间的首端，b 表示区间的长度。

同时因为rand产生的是伪随机数，所以需要srand(time(0))来播种，同时根据我遇到的问题，srand(time(0))不应该放在循环语句中。

其次，在很多时候，因为数组的下标是从0开始的，为了方便我们可以扩大数组，来达到让数组的下标和变量的下标相对应的目的，提高可读性，让程序更加自然，比如本程序中设置了一个大小为13的数组，但实际上a[0]和a[2]都是没有用的。

Follow up

2数组的最大下标应该是数组的大小-1（因为下标从0开始），并且在这种C风格下的基于指针的数组无法检查是否超出边界，一点超出数组的边界，就会得到一个不确定的结果，因此在使用C风格的数组的时候应该注意其边界。

3如果auto型的数组不进行初始化的话，那么数组的内容将是不确定的，也就是说像本程序这样得到错误的结论。但是static类型的数组不初始化，将被自动初始化为0

4在将多维数组传递给函数的时候又两种方式，第一种通过引用传递地址

但是要注意的是这种方式，如果要传递二维以上的数组，第二个以及以后的下标都不能省略，这是因为多维数组在内存中的储存方式决定的（用一维数组的方式来模拟多维数组）即 p=a+n\*i+j a表示第一个元素的首地址，p表示a[i][j]的地址，而n表示列数。从这个式子我们可以看到，列数是不能被省略的。