



链滴

# Jzxx 2029: 【入门】铺地砖

作者: [cms42](#)

原文链接: <https://ld246.com/article/1572604691171>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

# 题目

[点我查看原题](#)

## 题目描述

在一个 $N \times N$ 的正方形房间地面上铺 $M \times M$ 的正方形地砖，问一共需要多少块这样的地砖？数据保证用这的地砖能正好铺满房间地面。

## 输入

一行：两个整数 $N$ 和 $M$  ( $1 \leq M \leq N \leq 1000000$ ，且 $N \bmod M = 0$ )。

## 输出

一行：一个整数，表示一共用的地砖块数。数据保证输出的结果 $< 2147483647$ 。

## 样例输入

8 4

## 样例输出

4

## 提示

房间地面大小为 $8 \times 8$ ，地砖大小为 $4 \times 4$ ，  
所需的地砖块数 =  $(8 \times 8) \div (4 \times 4) = 4$ 块。

## 分析

这是一道看上去很简单的题目：读入 $N, M$ ，输出 $(N \times N) / (M \times M)$ ，搞定。  
于是你兴冲冲地码完了程序：

```
#include<iostream>
using namespace std;

int main(){
    int n,m;
    cin >> n >> m;
    cout << (n*n) / (m*m) << endl;
    return 0;
}
```

并且还过了样例！

于是你兴冲冲地按下了提交，于是收获了 **答案错误** 的好成绩。

答案错误

大佬走过来，告诉你：“把  $(n*n) / (m*m)$  改成  $(n / m) * (n / m)$  就能过了！”

你重新提交了代码，获得了梦寐以求的 **正确**。 正确

但是，为什么会这样呢？这两个式子在数学上不是等价的吗？

等等，记得老师讲过，一个变量就像一个杯子，你可以往里面装东西。众所周知，杯子不是无限大的，能装的东西总量是 **有限** 的，那么变量装东西的总量是不是也是有限的呢？不妨查一下资料。

哈，你猜对了！我们发现，INT的最大值是：`INT_MAX=2147483647`

有没有觉得这个数字很眼熟？

输出

一行：一个整数，表示一共用的地砖块数。 数据保证输出的结果<2147483647。

题目告诉我们， **答案 < 2147483647**，也就是  $(n * n) / (m * m) < 2147483647$ 。

但是  $1 \leq M \leq N \leq 1000000$ ，因此  $n * n$  很可能大于 **2147483647**！

这又会怎么样呢？我们知道，一个杯子装不下东西，多余的东西就会流出来，使桌子上一塌糊涂；那，一个变量装不下东西，多余的东西流出来会怎么样呢？

来，试一试计算  $1000000 * 1000000$ ：

```
cms@cms:~/01/01/online/oj.jzxx.net/2/0/2/2029: [入门] 铺地砖 $ ./a.out
-727379968
```

答案居然是 **-727379968**！

果然，在计算机中，“多余的东西”流出来也会造成大麻烦！（这称作“溢出”，是今后常常困扰我的问题）

我们的程序计算  $(n * n)$  的结果可能会使int溢出，产生错误的结果，怪不得我们被答案错误了！

那为什么大佬的程序就对了呢？因为题目保证答案小于 **2147483647**，那么  $(n / m)$  也必然小于 **2147483647**，也就不会溢出来了！

（提示：虽然题目保证最终答案不会溢出，但也要当心中间过程溢出）

那么我们可以怎么办呢？第一种办法就是像大佬一样，避免中间过程产生太大的结果。

那如果我们做不到呢？

如果你的杯子装不下水，除了少装一点，还有什么办法呢？

没错，换一个大杯子！

在C++中，我们有很多不同大小的杯子，列举一些常用的如下：

名称	字节长度	取值范围
bool	1	false 到 true
char、signed char	1	-128 到 127

unsigned char	1	0 到 255
int、long 483,648 到 2,147,483,647 (一般情况)	4 (一般情况)	-2,147
long long 775,808 到 9,223,372,036,854,775,807	8	-9,223,372,036,854

int对我们来说太小了，那么我们换成更大的long long不就行了吗？

以下给出代码：

```
#include<iostream>
using namespace std;
```

```
int main(){
    long long n,m;
    cin >> n >> m;
    cout << (n*n) / (m*m) << endl;
    return 0;
}
```

然后你就可以骄傲地告诉大佬：我还有一种方法也能做对！

作为拓展，我们再提一种方法：

```
#include<iostream>
using namespace std;
```

```
int main(){
    int n,m;
    cin >> n >> m;
    cout << ((long long)n*n) / ((long long)m*m) << endl;
    return 0;
}
```

这又是什么意思？

原来，这里的(long long)意思是强制类型转换，也就是告诉电脑：我的n和m是装在int型杯子里的，是你必须把它装到long long杯子里进行运算！

至此，我们完美地解决了这道题。

最后留一个思考题：如果我的东西连long long和更大的\_int128都装不下了，这可怎么办？