

7-7 Sum of Factorials (10 分)

这道题目是考察 for 循环语句的使用，按照题目意思模拟嵌套循环即可。

0-4: int

5-9: long long, 这里我卡了一下数据类型，希望大家日后仔细读题，先分析好了极限数据的数据范围，再去确定数据类型。显然当 $n=20$ 的时候，光 20 的阶乘 int 已经放不下了，那么就需要更大的数据范围。

10-10: 那么问题又来了，如果数据范围再大应该怎么办呢？如果 long long 都放不下的数据，就需要借助一点算法的技巧了。这最后一分我给的是高精度算法，是一个选做的挑战测试点不要求必须通过，感兴趣的同学可以自行百度搜索学习。这个题 1k 次提交只有 5% 的通过率，惨不忍睹...

C 和 C++ 语言都得手写高精度，如果以后学习了 Python 或者 java，这些都是自带高精度的语言

Python

Python整数基本运算就是高精度计算

```
1 a=eval(input())
2 b=eval(input())
3 print(a+b)
4 print(a-b)
5 print(a*b)
6 print(a/b)
7 #高精度整数
```

Java

Java内部写好了高精度运算，只需要使用BigInteger类或者BigDecimal类即可

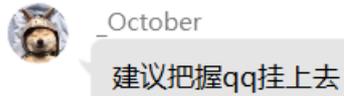
```
1 //高精度A+B
2 import java.util.Scanner;
3 import java.math.BigInteger;
4
5 public class plus {
6     private static Scanner in = new Scanner(System.in);
7     public static void main(String[] args) {
8         System.out.println(
9             in.nextBigInteger().add(in.nextBigInteger()) );
10    }
11 }
```

可能就会瞬间切了这道题。

标称链接: <https://paste.ubuntu.com/p/7yrngJKkXS/>

7-8 Number (5 分)

这是一道思维题，也是看起来挺难的一道题。非常帅气的_October 学长对此：



只有三百的提交还有不到 12%的通过率，看起来有点无从下手。

首先，先看一下数据范围， n 极限有 $1e3$ ， a_i 极限有 $1e6$ ，假设答案是两个极限的 a_i 相乘，那么答案就会是 $1e12$ ，显然又爆了 int 。所以这题还是需要开 $long long$ 。得了三分的同学估计就是没开 $long long$ 被卡了两分。我怕这个题被卡数据范围还特意在前面专门出了一道考察数据范围的 7-7 作为预防针。

然后再来分析题目：找到一个区间，使得区间里面的最大值和最小值乘积最大。

考虑一个区间三元组 a, b, c ，假设 $a * c$ 是答案， a 为最小， c 为最大，则一定满足 $a \leq b \leq c$ ，那么最优答案就会被更新为 $b * c$ 。所以，对于整个区间去考虑，那么最优的答案就是相邻的两两相乘取 \max 。这道题标称的核心代码只有 3 行，过了这个题的同学大部分写的两个 for 循环嵌套（我在出题的时候没想到还能有这种解法），这种也是可以的，只不过会慢一些，数据规模一大就被卡了。但看起来正解的代码更短更简单些...

标称链接：<https://paste.ubuntu.com/p/3rKGPB5sWy/>

7-9 _October's Saving Plan (5 分)

这道题...其实是一道很简单的小模拟，只不过题面比较长。考察对题意的理解与把握（通过出题人的 poor English 去猜测他想干嘛）和对问题的模拟能力。

_October 每个月能获得 300 块，他随时又可以去以整百的数额去存钱，然鹅他又要花费一定的钱，请问他是否一直拥有足够的钱，如果可以，会额外获得 20%，反之，输出钱不够的月份。

主要锻炼一下大家的英语和模拟能力.....貌似也没有什么坑点和难点，显然这个题在 int 范围内，所以不必担心开 long long，可能因为_October 他只有六岁刚开始上小学所以并没有 9 位数的资产，说不定以后就有了呢/

标称链接（代码注释写的很清楚）：

<https://paste.ubuntu.com/p/ycqNKcVnZy/>