

模拟赛 1(C 组)

题目名称	巨型整数	排试场	打水漂	合并排列数
存盘文件名	hugeint.cpp	move.cpp	watfly.cpp	num.cpp
输入文件名	hugeint.in	move.in	watfly.in	num.in
输出文件名	hugeint.out	move.out	watfly.out	num.out
时限	1s	1s	1s	1s
内存限制	64M	64M	64M	64M

【注意事项】：请自行完成题目，切勿讨论。

第 1 题 巨型整数

【问题描述】

给定一个任意整数，请您对它作出如下处理，从第 1 位开始读到最后一位，然后从最后一位开始倒序输出，再请您求出各个位数上的和。{够简单吧~~~}

【输入格式】 hugeint.in

一个任意整数 n 。对于 20% 的数据 $n \leq 9 \times 10^9$ ，对于 40% 的数据 $n \leq 9 \times 10^{200}$ ，对于 60% 的数据 $n \leq 9 \times 10^{2000}$ ，对于 100% 的数据 $n \leq 9 \times 10^{20000}$

【输出格式】 hugeint.out

输出文件为两行，第一行为处理后得到的新的整数，第二行为所求得各个位数上的和

【输入样例】

2516

【输出样例】

6152

14

“C++语言” 程序格式:

```
#include<stdio>
int main()
{
    freopen("hugeint.in", "r", stdin);
    freopen("hugeint.out", "w", stdout);
    /* 中间按原样写代码，什么都不用修改 */
    fclose(stdin);fclose(stdout);
    return 0;
}
```

“Pascal 语言” 程序格式:

```
begin      // 主程序开始
    assign(input, 'hugeint.in');reset(input);
    assign(output, 'hugeint.out');rewrite(output);
    /* 中间按原样写代码，什么都不用修改 */
    close(input);close(output);
end.      //主程序结束
```

第 2 题 排试场(move)

【问题描述】

期中考之前，总要排一下试场。校长为此心烦——每个试场拥有的桌椅数不同。为了排试场，不免有些桌子要被移到别的试场去。为了减少移动量，校长决定再开 M 个试场 (M 尽量要小，这是人之常情)，将每个试场多余的桌椅放入新试场内。由于教育局规定：**每个试场的桌椅数必须相同**。校长犯难了，于是他找到学 NOIP 的你，希望你能帮他点忙。(原试场是只出不进，新加的试场只进不出,)。如下图：原先有了 4 个试场 (四个角上，分别有 11、12、13、14 张桌椅)，在中央加入了一个试场，从第一个试场 (左上角) 取 1 套桌椅，从第二个试场 (左下角) 取 2 套桌椅，从第三个试场 (右下角) 取 3 套桌椅，从第四个试场 (右上角) 取 4 套桌椅，移入中央的试场，则第一个试场还有 $11-1=10$ 套桌椅，第二个试场还有 $12-2=10$ 套桌椅，第三个试场还有 $13-3=10$ 套桌椅，第四个试场还有 $14-4=10$ 套桌椅，新加的中央试场就有了 $1+2+3+4=10$ 套桌椅。这时，每个试场都有 10 套桌椅，就符合教育局的要求 (每个试场的桌椅数必须相同)。

【输入格式】

第一行两个数 n ($n \leq 10000$)， r ($r \leq 60$)，其中 n 代表试场数， r 表示校长希望每个试场能有的桌椅数。注：给出的数据一定有解。

接下来 n 行，每行一个数 x ，第 i 行表示第 i 个试场所拥有的桌椅数 x ($x \leq 80$)

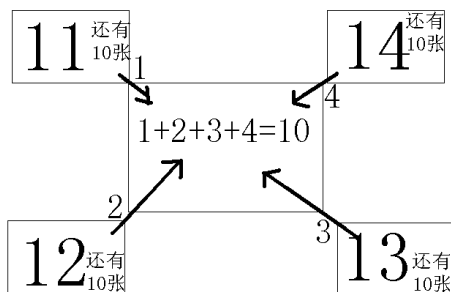
【输出格式】

第一行：用一个空格隔开的两个数 m, p 。 m 表示需要再开的试场数 (可以不开，不开时为 0， m 为正整数)， p 表示每个试场的桌椅数 (p 也为正整数)。

第二行：一个字符 'T' 或 'F' (必须是大写)，表示 p 是否大于 r ，若 $p > r$ 输出 'T'，否则，则输出 'F'。

【输入样例】

```
5 10
22
28
23
27
25
```



【输出样例】

```
20 5
F
```

“C++语言” 程序格式:

```
#include<cstdio>
int main()
{
    freopen("move.in", "r", stdin);
    freopen("move.out", "w", stdout);
    /* 中间按原样写代码，什么都不用修改 */
    fclose(stdin);fclose(stdout);
    return 0;
}
```

第3题 打水漂

【问题描述】

君不知，打靶大牛 goleenuoer 可喜欢打水漂了，他的靶子可以打到河面上的任何一条鱼，可是他的水漂打得实在是烂，无论怎么打那石子只会在河面上跳跃几次就“扑通”了。这天他又来打了，这条宽 w 米，每隔一米都会有一条鱼，每条鱼都有它的美观值。他想知道如何打才能得到两条鱼之间最大的美观值总和，刚接触 OI 的他想请您来解答，您能帮助他吗？

【输入格式】 watfly.in

第一行为一个整数 n ($n \leq 10000$)，从第二行起 n 个数，第 i 个整数表示第 i 条鱼的美观值范围 ($-500..500$)，当所有整数都为负数时输出 0。

【输出格式】 watfly.out

输出文件包含两行，第一行为石子的起点和落点，用空格隔开。第二行为一个整数表示所得到的两条鱼之间美观值总和。

【输入样例】

```
6
-2 11 -4 13 -5 -2
```

【输出样例】

```
2 4
20
```

“C++语言” 程序格式：

```
#include<cstdio>
int main()
{
    freopen("watfly.in", "r", stdin);
    freopen("watfly.out", "w", stdout);
    /* 中间按原样写代码，什么都不用修改 */
    fclose(stdin);fclose(stdout);
    return 0;
}
```

第4题 合并排列数

【问题描述】

tttak 自上了初中以后，就很喜欢钻研各种理科知识，他在假期学习了高中数学中的排列组合，主要是为了在 OI 比赛中起到一些作用。

之后，他一直想用所学的知识表达出一些奇异的想法，经过几个星期的思考，他终于想出了一个关于排列的特殊数，即合并排列数。

关于这个数，tttak 是这样定义的：

我们已知 $A(m, n) = (n-m+1) * [(n-m+1)+1] * \{[(n-m+1)+1]+1\} * \dots * n$ 也就是从 n 个元素中选取 m 个元素排列生成不同序列的个数 ($m \leq n$, 且均为正整数)，例如 $A(2, 5) = 4 * 5 = 20$ 。但仅

仅是这样一个公式太单调了，于是我们可以把生成的序列先按照顺序列出来，即 $A(2, 5)$ 所生成的序列为 $\{1, 2\} \{1, 3\} \{1, 4\} \{1, 5\} \{2, 1\} \dots \{5, 4\}$ ，然后再按照一个排列的元素顺序进行合并，则合并之后得到的新数即为 $\{12\} \{13\} \{14\} \{15\} \{21\} \dots \{54\}$ ，现在我们将其中的合数全部去掉，则还有如下合并排列数 $\{13\} \{23\} \{31\} \{41\} \{43\} \{53\}$ ，然后再算出所有剩下数的乘积，得到的数就叫做合并排列数。

现在请你根据 ttak 设计的这个方法，算出 $A(m, n)$ 的合并排列数。

【输入格式】

只有一行，即 $A(m, n)$ 中的 m 和 n 。

【输出格式】

只有一行，即得到的合并排列数。

【输入样例】

2 5

【输出样例】

866086091

【数据规模】

对于 80% 的数据， $0 < m \leq n \leq 4$ ；

对于 100% 的数据， $0 < m \leq n \leq 5$ 。

“C++语言” 程序格式：

```
#include<cstdio>
int main()
{
    freopen("num.in", "r", stdin);
    freopen("num.out", "w", stdout);
    /* 中间按原样写代码，什么都不用修改 */
    fclose(stdin);fclose(stdout);
    return 0;
}
```