

本题可使用本地IDE编码，不做跳出限制，编码后请点击“保存并调试”按钮进行代码提交。

■ 题目描述

读入一个数列和N值，返回按优先级排序的N个数，满足：

- (1) 所有偶数优先级大于奇数
- (2) 同为偶数或同为奇数时，数值大的优先级高

输入描述:

每个测试输入的测试用例，包含一个用半角逗号(,)分开的自然数数列和1个参数N，数列和参数N用半角分号(;)隔开。

这里保证N小于数列的元素个数(不超过100)。

输出描述:

在一行内输出N个满足题目条件的自然数，用逗号隔开

示例1 输入输出示例仅供调试，后台判题数据一般不包含示例

输入

复制

```
555503,772867,756893,339138,399447,40662,859037,628085,855723,97
4471,599631,636153,581541,174761,948135,411485,554033,858627,402
```

收起答题卡 ▾

☒ 已答 ☐ 未答 ☐ 当前

示例1 输入输出示例仅供调试，后台判题数据一般不包含示例

输入

复制

```
555503,772867,756893,339138,399447,40662,859037,628085,855723,97
4471,599631,636153,581541,174761,948135,411485,554033,858627,402
833,546467,574367,360461,566480,755523,222921,164287,420256,4004
3,977167,543295,944841,917125,331763,188173,353275,175757,950417
,284578,617621,546561,913416,258741,260569,630583,252845;10
```

输出

复制

```
913416,566480,420256,339138,284578,40662,977167,974471,950417,94
8135
```

■ 题目描述

产品经理小梅喜欢和他的男朋友小白一块玩扑克游戏。每一局，小梅抽取 N 张扑克牌，自左向右依次排列在桌面上；小白也抽取 M ($8 \geq N \geq M \geq 1$) 张扑克牌，自左向右依次排列在桌面上。

小梅需要进行 N 个回合，使用手中的扑克牌，组成一个新的扑克牌的序列。每个回合，小梅有 d 、 l 、 r 三种策略：

- 选择 d 时，小梅将最左边的扑克牌丢弃
- 选择 l 时，小梅将最左边的扑克牌取出，放到新的扑克牌序列的最左边
- 选择 r 时，小梅将最左边的扑克牌取出，放到新的扑克牌序列的最右边

N 个回合完成，新的扑克牌序列与小白的扑克牌完全一样（只考虑数字，不考虑花色），则小梅胜出。

小梅向程序员小岛提了一个需求，希望了解获胜的全部方法。简化起见，扑克牌仅包含1-9。

输入描述:

首先，输入数字 S ，作为局数 ($1 \leq S \leq 10$)。

每一局，分别输入两行字符串，分别代表小梅的抽取的扑克牌（从左向右排列）和小白抽取到的扑克牌（从左向右排列）。

输出描述:

对于每一局，在开始和结束，分别输出 { 和 }。

输出获胜的方法，回合策略的结尾输出一个空格。若存在多个获胜方法，请按字典序的升序输出。

示例1 输入输出示例仅供调试，后台判题数据一般不包含示例

示例1 输入输出示例仅供调试，后台判题数据一般不包含示例

输入

复制

```
3
123
3
123
321
45
67
```

输出

复制

```
{
d d l
d d r
}
{
l l l
r l l
}
{
}
```

扔 n 个骰子，第 i 个骰子有可能投掷出 X_i 种等概率的不同的结果，数字从1到 X_i 。所有骰子的结果的最大值将作为最终结果。求最终结果的期望。

输入描述:

第一行一个整数 n ，表示有 n 个骰子。 ($1 \leq n \leq 50$)
第二行 n 个整数，表示每个骰子的结果数 X_i 。 ($2 \leq X_i \leq 50$)

输出描述:

输出最终结果的期望，保留两位小数。

示例1 输入输出示例仅供调试，后台判题数据一般不包含示例

输入

复制

```
2
2 2
```

输出

复制

```
1.75
```

[编程|25分] 二维表第k大数

时间限制：C/C++ 1秒，其他语言 2秒

空间限制：C/C++ 32768K，其他语言 65536K

64bit IO Format: %lld

本题可使用本地IDE编码，不做跳出限制，编码后请点击“保存并调试”按钮进行代码提交。

■ 题目描述

在一块长为 n ，宽为 m 的场地上，有 $n \times m$ 个 1×1 的单元格。每个单元格上的数字就是按照从1到 n 和1到 m 中的数的乘积。具体如下

$n = 3, m = 3$

1 2 3

2 4 6

3 6 9

给出一个查询的值 k ，求出按照这个方式列举的的数中第 k 大的值 v 。

例如上面的例子里，

从大到小为(9, 6, 6, 4, 3, 3, 2, 2, 1)

$k = 1, v = 9$

$k = 2, v = 6$

$k = 3, v = 6$

...

$k = 8, v = 2$

$k = 9, v = 1$

输入描述:

只有一行是3个数 n, m, k 表示场地的宽高和需要查询的 k 。使用空格隔开。

输出描述:

给出第 k 大的数的值。

输出描述:

给出第k大的数的值。

示例1 输入输出示例仅供调试，后台判题数据一般不包含示例

输入

复制

3 3 4

输出

复制

4

备注:

【数据范围】

100%的数据

$1 \leq n, m \leq 40000$

$1 \leq k \leq n * m$

30%的数据

$1 \leq n, m \leq 1000$

