

## 休闲模拟赛 Day2

zzq

2018.12

题目名称	选举	排序	排序
源程序文件名	elect.c/cpp/pas	sort.c/cpp/pas	sortb.c/cpp/pas
输入文件名	elect.in	sort.in	sortb.in
输出文件名	elect.out	sort.out	sortb.out
时间限制	1s	1s	1s
是否捆绑测试	是	是	是
内存限制	512MB	512MB	512MB
是否有部分分	是	否	否
题目类型	传统	传统	传统
是否有附加文件	否	否	否
编译开关	-O2 -std=c++11	-O2 -std=c++11	-O2 -std=c++11

注意：

- 1.AK 了不要 D 出题人，没 AK 也不要 D 出题人。
2. 按照剧本所有题目时限均为 std 运行最大点用时两倍以上。
3. 题目顺序与难度无关。
4. 如有更改，输入输出文件、时间限制、内存限制、编译开关以 oj 上的为准。

## Problem A. 选举 (elect.c/cpp/pas)

Input file: `elect.in`  
 Output file: `elect.out`  
 Time limit: 1 second  
 Memory limit: 512 megabytes

Byteland 即将开始下一轮总统选举，有  $n$  个选民和  $m$  个候选人。为了体现民主性，选举分为若干轮，每轮每个选民有一张选票，需要投给恰好一个候选人。每一轮选举结束后，会公布每个候选人的该轮得票数，如果这一轮不是第一轮，且每个选民都和上一轮投给了相同的候选人，那么说明选民们达成了一致，否则继续开始下一轮投票。

由于 Byteland 的选民都是机器人，他们独特的投票方法。每个选民有一个自己喜欢的候选人的非空列表，从前往后表示喜好度递减。例如如果选民 1 的列表是  $\{2, 3\}$ ，表示他最喜欢候选人 2，其次喜欢候选人 3，其他的候选人他不喜欢。

第一轮投票的时候，每个选民都会投给自己最喜欢的候选人。从第二轮开始的每一轮，每个选民会投给自己的喜欢列表中上一轮得票最多的候选人，如果有多个选择最喜欢的。

作为一个旁观者，zzq 想知道这场投票可能会进行多少轮。请给出一个满足  $n \leq 1000, m \leq 1000$  的可能情形，轮数越高你的得分也会更高，评分细则请看 **Notes** 一节。

### Input

没有输入。

### Output

第一行两个正整数  $n, m$ 。你需要保证  $n, m \leq 1000$ 。

接下来  $n$  行每行一个列表，表示第  $i$  个选民的喜欢列表。首先是列表的长度（正整数），接下来按顺序列出列表内的元素（不同的  $[1, m]$  的正整数）。

### Examples

<code>elect.in</code>	<code>elect.out</code>
	3 3 3 2 3 1 1 2 2 1 2

在这个输出中，第一轮选民会依次投给 2, 2, 1 号候选人，第二轮 3 号选民会改变主意，投给 2 号候选人，第三轮就没人改变主意了，所以一共进行了 3 轮投票。当然，你输出这个并不会得到任何分数。

### Notes

本题就 1 个测试点。若你的输出格式不正确，得 0 分。

否则设你输出中的情形进行了  $x$  轮投票，设  $y = \{5, 10, 20, 30, 45, 60, 90, 190, 240, 490\}$ ，若  $y$  中有  $a$  个不超过  $x$  的元素，则你得  $10a$  分。

## Problem B. 排序 (sort.c/cpp/pas)

Input file:            **sort.in**  
Output file:           **sort.out**  
Time limit:            1 second  
Memory limit:         512 megabytes

zzq 近日发明了一种最新的排序算法 **FastSort**，它的伪代码大致如下：

---

```
function FASTSORT( $a, n$ )  
   $cnt \leftarrow 0$   
  for  $i \leftarrow 1$  to  $n$  do  
    for  $j \leftarrow i + 1$  to  $n$  do  
      if  $a_j < a_i$  then  
         $tmp \leftarrow a_j$   
         $a_j \leftarrow a_i$   
         $a_i \leftarrow tmp$   
      end if  
       $cnt \leftarrow cnt + 1$   
    end for  
  end for  
  return  $a$   
end function
```

---

当然，这个伪代码里的  $cnt$  对排序来说没什么用，它只是用来指示这个算法的运行效果的。

zzq 很喜欢这个算法，于是他打算用它来给一个  $1, 2 \dots n$  的排列  $a$  排序。为了清楚地获取算法的运行过程，zzq 决定在每次  $cnt$  改变时记录下当前的  $cnt$  和  $a$  便于分析。

zzq 运行了这个算法之后就睡觉去了。第二天早上，zzq 发现昨天停电了，日志里只剩下了记录下的最后一个  $cnt$  和  $a$ 。zzq 不想再运行一遍算法了，于是他想让你确认一下记录是否正确。

一句话题意：对给定的  $1, 2 \dots n$  的排列  $a$  执行算法 **FastSort**，问当  $cnt$  刚刚变成输入中给定的值时的  $a$  序列。

### Input

第一行两个整数， $n$  和  $cnt$ 。

第二行  $n$  个整数，表示  $a_1, a_2 \dots a_n$ 。保证是一个  $1, 2 \dots n$  的排列。

### Output

一行  $n$  个整数，表示  $cnt$  恰好变成给定值时的  $a$  序列。

## Examples

sort.in	sort.out
5 3 4 3 2 5 1	2 4 3 5 1
9 16 1 9 8 2 3 7 5 4 6	1 2 8 9 3 7 5 4 6

## Notes

对于所有数据， $2 \leq n \leq 10^6, 1 \leq cnt \leq \frac{n(n-1)}{2}$ 。

Subtask 1 (10pts):  $cnt \leq 10^7$ 。

Subtask 2 (10pts):  $a_i = n + 1 - i$ 。

Subtask 3 (20pts): 数据生成方式为，先选定  $n \leq 100000$  和  $cnt$ ，并初始化  $a_i = i$  ( $\forall i \in [1, n]$ )，然后重复 10 次，每次在  $[1, n]$  中独立均匀随机两个不同的整数  $i$  和  $j$ ，并交换  $a_i$  和  $a_j$ 。

Subtask 4 (20pts):  $n \leq 50000$ 。

Subtask 5 (20pts):  $n \leq 200000$ 。

Subtask 6 (20pts): 无特殊限制。

由于输入输出量较大，下发文件中有 `io.cpp`，选手可以参考。注意评测时并不会附加该文件，如果你需要使用它可以直接拷贝其源代码。

## Problem C. 排序 (sortb.c/cpp/pas)

Input file: `sortb.in`  
 Output file: `sortb.out`  
 Time limit: 1 second  
 Memory limit: 512 megabytes

懒得写题目背景了，就不写了。

有一个  $0, 1 \dots n-1$  的排列  $p_1, p_2 \dots p_n$ ，如果  $p_i \oplus p_j \leq a$ （其中  $\oplus$  为按位异或），你就可以交换  $p_i$  和  $p_j$ 。你希望通过若干次操作把它排序。但是，你还没有定下  $a$  的值，你想知道能够成功排序情况下最小的非负整数  $a$ 。

为了让题目更难，有时候会对排列进行更新，每次更新是交换某对  $p_i$  和  $p_j$ 。在每次更新后你都需要输出最小的  $a$ 。

### Input

第一行输入  $n$  和  $m$ ， $m$  是更新的个数。

接下来  $m$  行，每行两个整数  $i, j$ ，表示交换  $p_i$  和  $p_j$ 。

### Output

每个更新后输出一行最小的  $a$ 。

### Examples

sortb.in	sortb.out
5 3	0
0 1 3 2 4	4
3 4	4
1 5	
2 5	

### Notes

对于所有数据， $1 \leq n, m \leq 500000$ ，对于每个更新  $1 \leq i, j \leq n$ ， $i \neq j$ 。

Subtask 1 (10pts):  $n, m \leq 5$ 。

Subtask 2 (20pts):  $n, m \leq 300$ 。

Subtask 3 (20pts):  $m = 1$ 。

Subtask 4 (20pts):  $n, m \leq 50000$ 。

Subtask 5 (30pts): 无特殊限制。