

随机数生成器

【问题描述】

小 H 最近在研究随机算法。随机算法往往需要通过调用随机数生成函数（例如 Pascal 中的 `random` 和 C/C++ 中的 `rand`）来获得随机性。事实上，随机数生成函数也并不是真正的“随机”，其一般都是利用某个算法计算得来的。

比如，下面这个二次多项式递推算法就是一个常用算法：

算法选定非负整数 x_0, a, b, c, d 作为随机种子，并采用如下递推公式进行计算。

$$\text{对于任意 } i \geq 1, \quad x_i = (a \cdot x_{i-1}^2 + b \cdot x_{i-1} + c) \bmod d$$

这样可以得到一个任意长度的非负整数数列 $\{x_i\}_{i \geq 1}$ ，一般来说，我们认为这个数列是随机的。

利用随机序列 $\{x_i\}_{i \geq 1}$ ，我们还可以采用如下算法来产生一个 1 到 K 的随机排列 $\{T_i\}_{i=1}^K$ ：

- 1、初始设 T 为 1 到 K 的递增序列；
- 2、对 T 进行 K 次交换，第 i 次交换，交换 T_i 和 $T_{(x_i \bmod i)+1}$ 的值。

此外，小 H 在这 K 次交换的基础上，又额外进行了 Q 次交换操作，对于第 i 次额外交换，小 H 会选定两个下标 u_i 和 v_i ，并交换 T_{u_i} 和 T_{v_i} 的值。

为了检验这个随机排列生成算法的实用性，小 H 设计了如下问题：

小 H 有一个 N 行 M 列的棋盘，她首先按照上述过程，通过 $N \times M + Q$ 次交换操作，生成了一个 $1 \sim N \times M$ 的随机排列 $\{T_i\}_{i=1}^{N \times M}$ ，然后将这 $N \times M$ 个数逐行逐列依次填入这个棋盘：也就是第 i 行第 j 列的格子上所填入的数应为 $T_{(i-1) \cdot M + j}$ 。

接着小 H 希望从棋盘的左上角，也就是第一行第一列的格子出发，每次向右走或者向下走，在不走出棋盘的前提下，走到棋盘的右下角，也就是第 N 行第 M 列的格子。

小 H 把所经过格子上的数字都记录了下来，并从小到大排序，这样，对于任何一条合法的移动路径，小 H 都可以得到一个长度为 $N + M - 1$ 的升序序列，我们称之为路径序列。

小 H 想知道，她可能得到的字典序最小的路径序列应该是怎样的呢？

【输入格式】

从文件 `random.in` 中读入数据。

输入文件的第 1 行包含 5 个整数，依次为 x_0, a, b, c, d ，描述小 H 采用的随机数生成算法所需的随机种子。

第 2 行包含三个整数 N, M, Q ，表示小 H 希望生成一个 1 到 $N \times M$ 的排列来填入她 N 行 M 列的棋盘，并且小 H 在初始的 $N \times M$ 次交换操作后，又进行了 Q 次额外的交换操作。

接下来 Q 行，第 i 行包含两个整数 u_i, v_i ，表示第 i 次额外交换操作将交换 T_{u_i}

和 T_{v_i} 的值。

【输出格式】

输出到文件 *random.out* 中。

输出一行，包含 $N + M - 1$ 个由空格隔开的正整数，表示可以得到的字典序最小的路径序列。

【样例输入 1】

```
1 3 5 1 71
3 4 3
1 7
9 9
4 9
```

【样例输出 1】

```
1 2 6 8 9 12
```

【样例输入 2】

```
654321 209 111 23 70000001
10 10 0
```

【样例输出 2】

```
1 3 7 10 14 15 16 21 23 30 44 52 55 70 72 88 94 95 97
```

【样例输入 3】

```
123456 137 701 101 10000007
20 20 0
```

【样例输出 3】

```
1 10 12 14 16 26 32 38 44 46 61 81 84 101 126 128 135
140 152 156 201 206 237 242 243 253 259 269 278 279 291 298
338 345 347 352 354 383 395
```

【样例说明】

对于样例 1，根据输入的随机种子，小 H 所得到的前 12 个随机数 x_i 为：

9 5 30 11 64 42 36 22 1 9 5 30

根据这 12 个随机数，小 H 在进行初始的 12 次交换操作后得到的排列为：

6 9 1 4 5 11 12 2 7 10 3 8

在进行额外的 3 次交换操作之后，小 H 得到的最终的随机排列为：

12 9 1 7 5 11 6 2 4 10 3 8

这个随机排列可以得到如右侧的棋盘：

最优路径依次经过的数字为：
12→9→1→6→2→8。

12	9	1	7
5	11	6	2
4	10	3	8

对于样例 3，由于卷面宽度不够，在样例输出中出现了换行。请注意，这里的换行仅作展示用途，事实上，样例输出有且仅有一行，所有的数字都应该出现在同一行中。

【样例输入输出 4】

见选手目录下的 *random/random.in* 与 *random/random.ans*。

【数据规模与约定】

所有测试数据的范围和特点如下表所示

测试点编号	N, M 的规模	Q 的规模	约定
1	$2 \leq N, M \leq 8$	$Q = 0$	$0 \leq a \leq 300$
2	$2 \leq N, M \leq 200$		
3			
4	$2 \leq N, M \leq 2000$	$0 \leq Q \leq 50000$	$0 \leq b, c \leq 10^8$
5			
6			
7	$2 \leq N, M \leq 5000$		$0 \leq x_0 < d \leq 10^8$
8			
9			
10			

【特别提示】

本题的空间限制是 256 MB，请务必保证提交的代码运行时所使用的总内存空间不超过此限制。

一个 32 位整数（例如 C/C++ 中的 int 和 Pascal 中的 Longint）为 4 字节，因而在程序中声明一个长度为 1024×1024 的 32 位整型变量的数组，将会占用 4 MB 的内存空间。