

常青数编程 2 月普及组月赛 & yLOI2023

LGR-131 yLOI2023

时间：2023 年 2 月 11 日 14:30~18:00

题目名称	分解只因数	苦竹林	云梦谣	腐草为萤
目录	ibv1	ring	dream	glowworm
题目类型	传统型	传统型	传统型	传统型
可执行文件名	ibv1	ring	dream	glowworm
输入文件名	标准输入	标准输入	标准输入	标准输入
输出文件名	标准输出	标准输出	标准输出	标准输出
每个测试点时限	1.0 秒	1.0 秒	2.0 秒	2.0 秒
内存限制	512 MiB	512 MiB	512 MiB	512 MiB
子任务数目	10	10	20	20
测试点是否等分	是	是	是	是

提交源程序文件名

对于 C++ 语言	ibv1.cpp	ring.cpp	dream.cpp	glowworm.cpp
-----------	----------	----------	-----------	--------------

编译选项

对于 C++ 语言	-O2 -lm -std=c++14
-----------	--------------------

【注意事项（请仔细阅读）】

1. 感谢常青数编程对本次比赛的大力支持。
2. 本 pdf 仅供参考，题面如有修改，以洛谷网页上的题面为准。
3. C++ 中函数 main() 的返回值类型必须是 int，程序正常结束时的返回值必须为 0。
4. 程序可使用的栈空间大小与该题内存空间限制一致。
5. 若无特殊说明，每道题的代码大小限制为 50KiB，输入与输出中同一行的相邻整数、字符串等均使用一个空格分隔，结果比较方式为全文比较（忽略行末空格、文末回车）。
6. 统一评测时采用的机器配置为 Intel(R) Xeon(R) Platinum 8369HC CPU @ 3.30GHz，内存 24GB，上述时限以此配置为准。
7. 评测在 Linux 下进行。数据与附加样例文件均为 Linux 格式的。
8. 选手不得使用 SSH 等方式在赛时违规获取其他选手源程序。
9. 请使用 dev-c++、写字板等程序打开附加样例文件，使用记事本打开可能会显示异常。

分解只因数 (ibv1)

【题目背景】

本次比赛由 yLOI 命题组命题。

对于较大的附加样例文件，请使用文件输入测试所提供的样例，不要将之复制到命令行。且提交至洛谷时请删除文件输入语句。请使用 dev-c++、写字板等程序打开附加样例文件，使用记事本打开可能会显示异常。

我们提供了“数据规模与约定”信息，当你发现无法完整地解决题目提出的问题时，你可以选择一个你可以解决的数据规模并完成代码，从而获得部分分数。切记不要执着于完全解决某道题而错过了获取其他题目更多分数的机会。祝好。

【题目描述】

给定一个正整数 n ，对 n 分解质因数。

设 $n = p_1 \times p_2 \times p_3 \times \cdots \times p_x$ ，其中 p_i 均为质数。

如果 p_i 均为奇数，则称 n 为『只因数』。

现在，给出若干个 n ，请你判断 n 是不是『只因数』。

【输入格式】

从标准输入读入数据。

本题单测试点内有多组测试数据。

第一行是一个整数，表示数据组数 T 。

接下来 T 行，每行一个整数，表示一组数据的 n 。

【输出格式】

输出到标准输出中。

对每组数据，输出一行一个字符串。如果 n 是『只因数』，请输出 “Yes”，否则输出 “No”（均不含引号）。

【样例 1 输入】

```
5
2
3
4
6
9
```

【样例 1 输出】

```
No
Yes
No
No
Yes
```

【样例 2】

见选手目录下的 *ibvl/ibvl2.in* 和 *ibvl/ibvl2.ans*。

【数据规模与约定】

本题共有 10 个测试点。对第 i 个测试点， $T = \max(1, i - 1)$ 。

- 对 30% 的数据， $n \leq 3$ 。
- 对 50% 的数据， $n \leq 10$ 。
- 另有 10% 的数据， n 为奇数。
- 另有 10% 的数据， n 为偶数。
- 对 90% 的数据， $n \leq 10^9$ 。
- 对 100% 的数据， $1 \leq T \leq 9$ ， $2 \leq n \leq 10^{18}$ 。

苦竹林 (ring)

【题目背景】

悬挂在屋檐下的风铃，摇晃的声音很动听。
思念就像梅雨下不停，我的心境一片泥泞。
散落在天际里的繁星，闪烁着你我的宿命。
当枫叶轻盈落入湖心，近看山水一片宁静。
——银临 & 涵昱《苦竹林》

【题目描述】

共有 n 个风铃悬挂在屋檐下，每个风铃都能发出一定音调的声音。从左到右给风铃从 1 至 n 编号，第 i 个风铃的音调是 a_i 。

为了表达内心的思念，扶苏决定在 n 个的风铃中取出 m 个，送给远方的朋友。

请你找到最小的整数 ε ，使得存在一种方案，能够从 n 个风铃中挑出 m 个，设挑出风铃的音调为 b_1, b_2, \dots, b_m ，满足对任意的 $1 \leq i, j \leq m$ ，都有 $|b_i - b_j| \leq \varepsilon$ 。

【输入格式】

从标准输入读入数据。

第一行是两个整数，表示风铃的个数 n 和挑选出风铃的个数 m 。

第二行有 n 个整数，表示每个风铃的音调。第 i 个整数表示 a_i 。

【输出格式】

输出到标准输出中。

输出一行一个整数，表示最小的 ε 。

【样例 1 输入】

```
5 3
1 2 3 4 5
```

【样例 1 输出】

```
2
```

【样例 2 输入】

```
6 4
1 7 8 3 4 6
```

【样例 2 输出】

```
4
```

【样例 2 解释】

一种选择的方案是选择第 2, 4, 5, 6 四个风铃，音调依次为 7, 3, 4, 6。可以得到对任何的 $1 \leq i, j \leq 4$ ，都有 $|b_i - b_j| \leq 4$ 。

另一种方案是选择第 2, 3, 5, 6 四个风铃，同样计算得到的 ε 为 4。

【样例 3】

见选手目录下的 *ring/ring3.in* 与 *ring/ring3.ans*。

【数据规模与约定】

- 对 10% 的数据， $m = 2$ 。
- 另有 10% 的数据， $m = n$ 。
- 对 40% 的数据， $n \leq 5$ 。
- 对 60% 的数据，保证对所有的 $2 \leq i \leq n$ ，满足 $a_{i-1} \leq a_i$ ，即 a_i 单调不降。
- 对 80% 的数据， $n \leq 10^3$ 。
- 对 100% 的数据， $2 \leq m \leq n \leq 10^5$ ， $1 \leq a_i \leq 10^9$ 。

云梦谣 (dream)

【题目背景】

归来且做云梦，梦一场大梦好，
栽花闻酒香，醒醒醉醉笑笑。
天地偌大复路远山高，最难得偷半日逍遥。
偶尔糊涂不问世事不知晓。
——银临 & 慕寒《云梦谣》

【题目描述】

“喂，枸杞，你这只笨狗，又偷吃！看我不收拾你！”

朵一气呼呼地从院子里跑出来，手中握着掸子，而枸杞早已不见踪影。

云梦庭可以看作一个 n 行 m 列的方格阵，第 i 行第 j 列的格子被记作 (i, j) 。每个格子 (i, j) 要么有一个高度 $h_{i,j}$ ($h_{i,j}$ 为正整数)，要么是障碍物，不能通过。（方便起见，约定障碍物的 $h_{i,j}$ 用 0 表示）另外，云梦庭上有 k 个指定的格子上可以进行御剑飞行。开始时，朵一和枸杞分别位于方格 $(1, 1)$ 和 (n, m) 。

朵一的御剑飞行还不是很熟练，现在她还控制不好御剑的高度。因此在任意时刻，朵一在方格 (i, j) 上可以做如下行动之一：

- 移动到与该方格相邻的方格 $(i-1, j)$ 、 $(i+1, j)$ 、 $(i, j-1)$ 、 $(i, j+1)$ 之一上（不能移动出方格边界，也不能移动到障碍物上）；
- 如果方格 (i, j) 上允许御剑飞行，则朵一可以御剑飞行至另一个同样允许御剑飞行且与方格 (i, j) 高度相等的方格上；
- 使用仙法将当前格子的高度 $h_{i,j}$ 改变为任一正整数。

进行上述每项行动均需花费 1 个单位时间。

“哼，笨狗子你再跑！”说罢，朵一便追了出去。朵一接下来还要尽快继续今天的修行，因此她想知道到达 (n, m) 格子所需的最短时间是多少。

【输入格式】

从标准输入中读入数据。

输入的第一行有三个整数，依次表示方格阵的行数 n 、列数 m 和有传送阵的方格个数 k 。

接下来 n 行，每行 m 个整数，其中第 i 行的第 j 个数表示方格 (i, j) 的高度 $h_{i,j}$ 。数据保证 $h_{1,1}$ 和 $h_{n,m}$ 不为 0。

接下来 k 行，每行两个整数 x 和 y ，表示一个允许御剑飞行的方格的坐标 (x, y) 。数据保证这 k 个方格的坐标互不相同。

【输出格式】

输出到标准输出中。

输出一行一个整数，表示朵一到达 (n, m) 所需的最小时间。如果朵一无法到达，输出 -1 。

【样例 1 输入】

```
4 4 2
1 2 3 4
1 2 3 4
1 2 3 4
1 2 3 4
1 1
3 4
```

【样例 1 输出】

```
3
```

【样例 1 解释】

第 1 个单位时间，朵一将当前方格 $(1, 1)$ 的高度修改为 4；

第 2 个单位时间，朵一从方格 $(1, 1)$ 御剑飞行至 $(3, 4)$ ；

第 3 个单位时间，朵一从方格 $(3, 4)$ 移动到 $(4, 4)$ ，追上了枸杞。

【样例 2 输入】

```
4 4 3
1 2 3 4
1 2 3 4
1 2 3 4
1 2 3 4
1 1
2 4
4 1
```

【样例 2 输出】

4

【样例 2 解释】

第 1 个单位时间，朵一从方格 (1, 1) 御剑飞行至 (4, 1)；

第 2 个单位时间，朵一从方格 (4, 1) 移动到 (4, 2)；

第 3 个单位时间，朵一从方格 (4, 2) 移动到 (4, 3)；

第 4 个单位时间，朵一从方格 (4, 3) 移动到 (4, 4)，追上了枸杞。

【样例 3 输入】

```
2 5 0
1 0 3 3 4
2 3 4 0 5
```

【样例 3 输出】

7

【样例 4 输入】

```
4 4 3
1 1 1 0
1 1 0 1
1 0 1 1
0 1 1 1
1 1
2 1
3 3
```


【样例 4 输出】

3

【样例 5】

见选手目录下的 *dream/dream5.in* 与 *dream/dream5.ans*。

【数据规模与约定】

测试点	$n, m \leq$	$k \leq$	$h_{i,j} \leq$	测试点	$n, m \leq$	$k \leq$	$h_{i,j} \leq$
1	4	0	1	11	3000	0	1
2				12			
3		2		13		10	
4			16	14			9×10^6
5	50	0	1	15		9×10^6	1
6				16			
7		2500		17			
8				18			
9		2500	19	9×10^6			
10			20				

对全部的测试点，保证 $1 \leq n, m \leq 3 \times 10^3$ ， $0 \leq k, h_{i,j} \leq n \times m$ 。

【提示】

请注意大量数据读入对程序效率造成的影响，选择合适的读入方式，避免超时。

【后记】

不过，别看朵一现在一副生气的样子，可当她追上枸杞后，大抵是不舍得真的动手吧。“嘿，今日的修行结束后，该吃什么好呢？”在这飞瀑悬挂、翠竹怀抱的云梦庭中，修仙炼体，不羡尘嚣，应是这世上最逍遥的事了。

腐草为萤 (glowworm)

【题目背景】

于盛夏之末，入夜仍灼热。
又一场离合，开始凄恻。
是扇底闪躲，或雨水摧折。
哪里都值得，恋恋不舍。
——银临《腐草为萤》

【题目描述】

夜幕降临，在树林中的一条平直的小径上，萤火虫们受到夜晚的呼唤，纷纷外出行动。

将小径视作数轴，一开始，共计 n 只萤火虫在数轴的一些整点的整点上，从左到右依次标号为 $1 \sim n$ ，第 i 只萤火虫的初始坐标为 x_i 。每个萤火虫有不同的亮度值， i 号萤火虫的亮度为 a_i 。

在任意时刻，对任意存活的萤火虫 i ，它会按如下规则飞行：

- 在当前仍存活的萤火虫中，找到与 i 相邻的萤火虫（可能是一只或两只）中亮度最大的一只，记其编号为 j 。如果 $a_i < a_j$ ，则 i 会朝着 j 飞行，否则 i 留在原地。
- 这里两只萤火虫『相邻』的定义是：若两只萤火虫之间不存在任何仍存活的萤火虫，则它们相邻。
- 萤火虫飞行的速度均为每秒一个单位长度。

萤火虫生命短暂，当两只萤火虫相遇之时（即两个萤火虫中的坐标相同时），亮度值较低的萤火虫将耗尽生命，在小径上消失。显然，最后只会剩余 1 只萤火虫。对其余的每只萤火虫，请分别求出它们耗尽生命时的坐标。

【输入格式】

从标准输入中读入数据。

第一行是一个整数，表示萤火虫数量 n 。

第二行有 n 个整数，第 i 个整数表示标号为 i 的萤火虫初始坐标 x_i 。数据保证 x_i 单调递增。

第三行有 n 个整数，第 i 个整数表示标号为 i 的萤火虫的亮度值 a_i 。数据保证亮度值互不相同。

【输出格式】

输出到标准输出中。

输出一行 n 个以单个空格隔开的整数，第 i 个整数表示编号为 i 的萤火虫生命耗尽时的坐标。如果 i 号萤火虫最后存活下来了，则第 i 个数输出 0。

【样例 1 输入】

```
4
1 2 3 4
2 3 1 4
```

【样例 1 输出】

```
2 4 4 0
```

【样例 1 解释】

在第一秒时，标号为 1 的萤火虫向右移动，标号为 2 的萤火虫位置不变，标号为 3 的萤火虫向右移动，标号为 4 的萤火虫位置不变。

第二秒开始时，萤火虫 1 遇到萤火虫 2，前者亮度更低，耗尽生命，此时其坐标为 2；萤火虫 3 遇到萤火虫 4，前者亮度更低，耗尽生命，此时其坐标为 4。

接下来，萤火虫 2 向右移动，直到在坐标 4 遇到萤火虫 4，耗尽生命。

【样例 2 输入】

```
5
1 2 3 4 5
5 3 2 1 4
```

【样例 2 输出】

```
0 1 1 5 1
```

【样例 3 输入】

```
5
2 4 6 10 12
5 3 1 4 2
```

【样例 3 输出】

```
0 2 2 2 10
```

【样例 4 输入】

```
7
2 4 6 8 12 14 16
5 3 2 6 1 4 7
```

【样例 4 输出】

```
8 2 8 16 16 16 0
```

【样例 5 输入】

```
7
2 4 6 8 12 14 16
7 1 6 3 5 4 2
```

【样例 5 输出】

```
0 2 2 6 2 12 2
```

【样例 6】

见选手目录下的 *glowworm/glowworm6.in* 与 *glowworm/glowworm6.ans*。

【样例 7】

见选手目录下的 *glowworm/glowworm7.in* 与 *glowworm/glowworm7.ans*。

【数据规模与约定】

- 对于 5% 的数据, $n = 2$ 。
- 对于 30% 的数据, $n \leq 100$, $x_i \leq 200$ 。
- 对于 60% 的数据, $n \leq 10^3$ 。
- 另有 5% 的数据, 满足特殊约定 A。
- 另有 5% 的数据, 满足特殊约定 B。
- 对 100% 的数据, 保证 $2 \leq n \leq 5 \times 10^5$, $1 \leq x_i \leq 10^9$, $1 \leq a_i \leq n$ 。且 $x_i < x_{i+1}$, a_i 是长度为 n 的排列。

其中:

- 特殊约定 A: 数列 a 单调递增。
- 特殊约定 B: 数列 a 是单峰的, 仅有一个极大值。即: 存在 p 满足 $1 \leq p < n$, 使得 $a_1 \sim a_p$ 单调递增, $a_p \sim a_n$ 单调递减。

【提示】

请注意大量数据的读入输出对程序效率造成的影响, 选择合适的读入输出方式, 避免超时。

请注意时间复杂度的常数因子对程序运行效率造成的影响。