

test

2025年1月23日 8点10分~12点40分

题目名称	信号传递	IIIDX	守卫
每个测试点时限	3s	2s	2s
内存限制	512MB	512MB	512MB
提交源程序文件名	transfer.cpp	data.cpp	guard.cpp

编译选项: -O2 -std=c++14 -static

信号传递

题目描述

一条道路上从左至右排列着 m 个信号站，初始时从左至右依次编号为 $1, 2, \dots, m$ ，相邻信号站之间相隔 1 单位长度。每个信号站只能往它右侧的任意信号站传输信号（称为普通传递），每单位长度距离需要消耗 1 单位时间。道路的最左侧有一个控制塔，它在最左侧信号站的左侧，与其相隔 1 单位长度。控制塔能与任意信号站进行双向信号传递（称为特殊传递），但每单位长度距离需要消耗 k 个单位时间。对于给定的长度为 n 的信号传递序列 S ，传递规则如下：

1. 共 $n - 1$ 次信号传递，第 i 次信号传递将把信号从 S_i 号信号站传递给 S_{i+1} 号。
2. 若 S_{i+1} 号信号站在 S_i 号右侧，则将使用普通传递方式，从 S_i 号直接传递给 S_{i+1} 号。
3. 若 S_{i+1} 号信号站在 S_i 号左侧，则将使用特殊传递方式，信号将从 S_i 号传递给控制塔，再由控制塔传递给 S_{i+1} 号。
4. 若 $S_i = S_{i+1}$ ，则信号无须传递。

阿基作为大工程师，他能够任意多次交换任意两个信号站的位置，即他能够重排信号站的顺序，这样会使得 S 消耗的传递时间改变。现在阿基想知道，在他重排信号站顺序后， S 所消耗的传递时间最小能是多少。

输入格式

第一行三个整数 n, m, k ，分别代表信号传递序列 S 的长度，信号站个数，特殊传递单位长度距离的时间消耗。

第二行 n 个整数，第 i 个整数表示 S_i 。

输出格式

仅一行一个整数表示答案。

样例 #1

样例输入 #1

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

样例输出 #1

```
2
```

样例 #2

样例输入 #2

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

样例输出 #2

```
6
```

提示

【样例解释 1】

信号站顺序保持不变，两次使用普通传递方式，时间消耗为 $1 + 1 = 2$ 。

【样例解释 2】

对于排列 1, 2, 3，传递时间为 $1 + 1 + (3 \times 1 + 1 \times 1) = 6$ 。

对于排列 1, 3, 2，传递时间为 $2 + (3 \times 1 + 2 \times 1) + (2 \times 1 + 1 \times 1) = 10$ 。

对于排列 2, 1, 3，传递时间为 $(2 \times 1 + 1 \times 1) + 2 + (3 \times 1 + 2 \times 1) = 10$ 。

对于排列 2, 3, 1, 传递时间为 $(3 \times 1 + 1 \times 1) + 1 + 1 = 6$ 。

对于排列 3, 1, 2, 传递时间为 $1 + (3 \times 1 + 1 \times 1) + 1 = 6$ 。

对于排列 3, 2, 1, 传递时间为 $(3 \times 1 + 2 \times 1) + (2 \times 1 + 1 \times 1) + 2 = 10$ 。

【数据范围】

30% 的数据： $m \leq 8, n \leq 100$ 。

60% 的数据： $m \leq 20$ 。

70% 的数据： $m \leq 21$ 。

80% 的数据： $m \leq 22$ 。

100% 的数据： $2 \leq m \leq 23, 2 \leq n \leq 10^5, 1 \leq k \leq 100, 1 \leq S_i \leq m$ 。

IIIDX

题目背景

Osu 听过没？那是 Konano 最喜欢的一款音乐游戏，而他的梦想就是有一天自己也能做个独特酷炫的音乐游戏。现在，他在世界知名游戏公司 KONMAI 内工作，离他的梦想也越来越近了。

这款音乐游戏内一般都包含了许多歌曲，歌曲越多，玩家越不易玩腻。同时，为了使玩家在游戏上氪更多的金钱花更多的时间，游戏一开始一般都不会将所有曲目公开，有些曲目你需要通关某首特定歌曲才会解锁，而且越晚解锁的曲目难度越高。

题目描述

这一天，Konano 接到了一个任务，他需要给正在制作中的游戏《IIIDX》安排曲目的解锁顺序。游戏内共有 n 首曲目，每首曲目都会有一个难度 d ，游戏内第 i 首曲目会在玩家 Pass 第 $\lfloor \frac{i}{k} \rfloor$ 首曲目后解锁（ $\lfloor x \rfloor$ 为下取整符号）若 $\lfloor \frac{i}{k} \rfloor = 0$ ，则说明这首曲目**无需解锁**。

举个例子：当 $k = 2$ 时，第 1 首曲目是无需解锁的（ $\lfloor \frac{1}{2} \rfloor = 0$ ），第 7 首曲目需要玩家 Pass 第 $\lfloor \frac{7}{2} \rfloor = 3$ 首曲目才会被解锁。

Konano 的工作，便是安排这些曲目的顺序，使得每次解锁出的曲子的难度**不低于**作为条件需要玩家通关的曲子的难度，即使得确定顺序后的曲目的难度对于每个 i 满足 $d_i \geq d_{\lfloor \frac{i}{k} \rfloor}$ 。

当然这难不倒曾经在信息学竞赛摸鱼许久的 Konano。那假如是你，你会怎么解决这份任务呢？

输入格式

第 1 行 1 个正整数 n 和 1 个小数 k ， n 表示曲目数量， k 其含义如题所示。

第 2 行 n 个用空格隔开的正整数 d ，表示这 n 首曲目的难度。

输出格式

输出 1 行 n 个整数，按顺序输出安排完曲目顺序后第 i 首曲目的难度。

若有多解，则输出 d_1 **最大的**；若仍有多解，则输出 d_2 **最大的**，以此类推。

样例 #1

样例输入 #1

4 2.0
114 514 1919 810

样例输出 #1

114 810 514 1919

提示

测试点编号	n	k	d	特殊限制
1	$1 \leq n \leq 10$	$k = 2$	$1 \leq d \leq 100$	保证 d_i 互不相同
2	$1 \leq n \leq 10$	$k = 3$	$1 \leq d \leq 100$	保证 d_i 互不相同
3	$1 \leq n \leq 10$	$k = 1.1$	$1 \leq d \leq 100$	保证 d_i 互不相同
4	$1 \leq n \leq 10$	$k = n$	$1 \leq d \leq 100$	保证 d_i 互不相同
5	$1 \leq n \leq 10$	$1 < k \leq 100$	$1 \leq d \leq 100$	保证 d_i 互不相同
6	$1 \leq n \leq 10$	$1 < k \leq 100$	$1 \leq d \leq 100$	保证 d_i 互不相同
7	$1 \leq n \leq 2000$	$k = 2$	$1 \leq d \leq 10^9$	保证 d_i 互不相同
8	$1 \leq n \leq 2000$	$k = 2$	$1 \leq d \leq 10^9$	无
9	$1 \leq n \leq 2000$	$k = 3$	$1 \leq d \leq 10^9$	保证 d_i 互不相同
10	$1 \leq n \leq 2000$	$k = 3$	$1 \leq d \leq 10^9$	无
11	$1 \leq n \leq 2000$	$1 < k \leq 10^9$	$1 \leq d \leq 10^9$	保证 d_i 互不相同
12	$1 \leq n \leq 2000$	$1 < k \leq 10^9$	$1 \leq d \leq 10^9$	无
13	$1 \leq n \leq 500000$	$k = 2$	$1 \leq d \leq 10^9$	无
14	$1 \leq n \leq 500000$	$k = 3$	$1 \leq d \leq 10^9$	无

测试点编号	n	k	d	特殊限制
15	$1 \leq n \leq 500000$	$1 < k \leq 10^9$	$1 \leq d \leq 10^9$	保证 d_i 互不相同
16	$1 \leq n \leq 500000$	$1 < k \leq 10^9$	$1 \leq d \leq 10^9$	保证 d_i 互不相同
17	$1 \leq n \leq 500000$	$1 < k \leq 10^9$	$1 \leq d \leq 10^9$	无
18	$1 \leq n \leq 500000$	$1 < k \leq 10^9$	$1 \leq d \leq 10^9$	无
19	$1 \leq n \leq 500000$	$1 < k \leq 10^9$	$1 \leq d \leq 10^9$	无
20	$1 \leq n \leq 500000$	$1 < k \leq 10^9$	$1 \leq d \leq 10^9$	无

守卫

题目描述

这一天她去爬山，她的父亲为了她的安全，雇了一些保镖，让他们固定地呆在山的某些位置，来实时监视九条可怜，从而保护她。

具体来说，一座山可以描述为一条折线，折线的下方是岩石。这条折线有 n 个折点，每个折点上有一个亭子，第 i 个折点的坐标是 (i, h_i) 。九条可怜只可能会在亭子处玩耍，那些保镖也只会在此处监视可怜。

由于技术方面的原因，一个保镖只能监视所有他能看得到的，横坐标不超过他所在位置的亭子。我们称一个保镖能看到一个亭子 p ，当且仅当他所在的亭子 q 和 p 的连线不经过任何一块岩石。特别地，如果这条连线恰好经过了除了 p, q 以外的亭子，那么我们认为保镖看不到可怜。

雇佣保镖是一件很费钱的事情，可怜的父亲希望保镖越少越好。

可怜的父亲还希望得到详尽的雇佣保镖的方案，他知道有些亭子可能正在维修，他想对所有的 $1 \leq l \leq r \leq n$ 计算：如果事先已知了只有区间 $[l, r]$ 的亭子可以用来玩耍（和监视），那么最少需要多少个保镖，才能让 $[l, r]$ 中的每一个亭子都被监视到。

可怜的父亲已经得到了一个结果，他希望和你核实他的结果是否正确。

输入格式

第一行输入一个整数 n 表示亭子的数目。

接下来一行 n 个整数，第 i 个整数 h_i 表示第 i 个亭子的坐标是 (i, h_i) 。

输出格式

对所有的 $1 \leq l \leq r \leq n$ 计算：如果事先已知了可怜只会在 $[l, r]$ 这个区间的亭子里面玩耍，那么最少需要多少个保镖，才能让 $[l, r]$ 中的每一个亭子都被监视到。由于输出量太大，可怜的父亲只要你输出所有 $[l, r]$ 的答案的异或即可。

样例 #1

样例输入 #1

```
3
2 3 1
```

样例输出 #1

```
3
```

提示

样例解释

如果 $r - l + 1 \leq 2$, 那么答案显然是 1。

如果 $l = 1, r = n$, 那么答案是 2, 需要安排两个保镖在 $(2, 3), (3, 1)$ 两个位置监视可怜。

数据范围与提示

对于 30% 的数据, $n \leq 20$ 。

对于 70% 的数据, $n \leq 500$ 。

对于 100% 的数据, $n \leq 5000, 1 \leq h_i \leq 10^9$ 。