

CSP-S 2019 模拟赛

第一试

时间：2019 年 10 月 dd 日 hh:mm ~ hh:mm

AHSOFNU

luhong & PinkRabbit

试题名称	俄罗斯方块	缩进优化	看电影
输入文件名	tetris.in	tab.in	movie.in
输出文件名	tetris.out	tab.out	movie.out
每个测试点时限	1.0 秒	1.5 秒	1.0 秒
内存限制	512 MB	512 MB	512 MB
子任务数目	10	10	10
子任务是否等分	是	是	是

提交源程序文件名

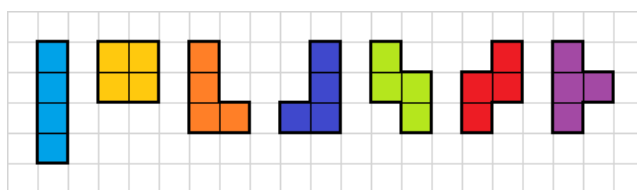
对于 C++ 语言	tetris.cpp	tab.cpp	movie.cpp
-----------	------------	---------	-----------

俄罗斯方块 (tetris)

【题目描述】

有一个高度为 n ，宽度为 2 的场地，你可以在其中放置若干俄罗斯方块。

共有 7 种俄罗斯方块，每种均可以旋转 0° 、 90° 、 180° 或 270° ：



请求出恰好铺满所有格子的放置方案数，两个方案不同当且仅当存在一个格子颜色不同。

【输入格式】

从文件 *tetris.in* 中读入数据。

本题输入文件包含多组数据。

每组数据仅一行一个正整数 n 。

【输出格式】

输出到文件 *tetris.out* 中。

对于每组数据输出一行一个数表示答案对 $10^9 + 7$ 取模的结果。

【样例输入】

2
4
5
6

【样例输出】

1
4
0
9

【数据范围与提示】

对于所有测试点： $1 \leq n \leq 10^{18}$ ，数据组数不超过 100。
每个测试点的具体限制见下表：

测试点编号	$n \leq$
1	10
2 ~ 3	1000
4 ~ 5	10^6
6 ~ 7	10^9
8 ~ 10	10^{18}

缩进优化 (tab)

【题目描述】

小 O 是一个热爱短代码的选手。在缩代码方面，他是一位身经百战的老手。世界各地的 OJ 上，很多题的最短解答排行榜都有他的身影。这令他感到十分愉悦。

最近，他突然发现，很多时候自己的程序明明看起来比别人的更短，实际代码量却更长。这令他感到很费解。经过一番研究，原来是因为他每一行的缩进都全是由空格组成的，大量的空格让代码量随之增大。

小 O 现在获得了 n 个不同的文本编辑器，每个文本编辑器有不同的 TAB 宽度，第 i 个编辑器以 a_i 个空格作为缩进。

编辑器的自动缩进是这样工作的：编辑器从头至尾不断把连续 x 个空格替换成一个 TAB，直到剩余空格数不足 x 个。最终缩进所占代码量为空格数与 TAB 数的和。

小 O 经常会更改某个编辑器的 TAB 宽度，或者询问一行代码在每个编辑器的自动缩进下的缩进所占代码量之和。具体地说，小 O 会进行 q 次操作，有如下两种操作：

- 1 p x: 更改第 p 个编辑器的 TAB 宽度，使得 a_p 改为 x 。
- 2 y: 询问一行以 y 个空格作为缩进的代码，在每个编辑器作用下的代码量之和。

【输入格式】

从文件 **tab.in** 中读入数据。

第一行两个正整数 n, q ，意义见题目描述。

第二行 n 个正整数 $a_{1\dots n}$ 依次表示每个编辑器的 TAB 宽度。

接下来 q 行，每行第一个数 op 表示操作种类。

若 op = 1，则接下来两个正整数 p, x 表示将 a_p 改为 x 。

若 op = 2，则接下来一个正整数 y 表示询问一行缩进为 y 的代码自动缩进后的代码量。

【输出格式】

输出到文件 *tab.out* 中。

对于每种 2 操作，输出一行一个数表示询问答案。

【样例 1 输入】

```
15 10
17 2 4 17 16 16 1 8 15 3 7 14 1 13 16
1 4 16
1 5 6
2 1
1 13 18
2 2
2 13
2 6
1 5 9
2 15
2 14
```

【样例 1 输出】

```
15
29
137
75
133
134
```

【样例 2】

见选手目录下的 *tab/tab2.in* 与 *tab/tab2.ans*。

【数据范围与提示】

对于所有测试点： $1 \leq n, q \leq 10^5$ ， $1 \leq p \leq n$ ， $1 \leq a_i, x, y \leq 2 \times 10^5$ 。

每个测试点的具体限制见下表：

测试点编号	$n \leq$	$q \leq$	特殊限制
1	2000	2000	无
2 ~ 3	10^5	2000	无
4 ~ 5	2000	10^5	无
6 ~ 7	10^5	10^5	没有 1 操作
8 ~ 10	10^5	10^5	无

看电影 (movie)

【题目描述】

《我和我的祖国》正在热映，共有 n 组小伙伴想要组团看电影！

影院座位可以看作一个 $k \times k$ 的正方形矩阵，且 k 是一个奇数。

越靠近中心的座位的观影体验越好，令 $x_c = y_c = \frac{k+1}{2}$ ，中心的座位即为 (x_c, y_c) 。

具体地说，一个座位 (x, y) 的观影体验被定义为 $|x - x_c| + |y - y_c|$ ，这个值越小观影体验越好。

假设第 i 组小伙伴有 m_i 个人，他们希望所有 m_i 个人均坐在同一排，且座位连续。假设排号为 x ，且座位号区间为 $[l, r]$ ，其中 $r - l + 1 = m_i$ ，则第 x 排的第 l 到第 r 个座位必须为空。如果不存在任意一行有 m_i 个连续的空座位，这组小伙伴就无法买到票，只能离开；否则他们会希望最小化如下距离函数值，这个函数值越小表示观影体验越好：

$$\sum_{i=l}^r |x - x_c| + |i - y_c|$$

若有多个可选座位均满足离影院中心距离最小的条件，则选座程序优先选择靠前的座位（即排号 x 最小的座位）。若仍有多个座位符合要求，则选座程序优先选择靠左的座位（即座位号 l 值最小的座位）。

假设电影院最开始没有售出任何座位，请问对于给出的 n 个购票请求，每次售出的票都能买到哪些座位？

【输入格式】

从文件 `movie.in` 中读入数据。

第一行两个正整数 n 和 k 表示购票请求的数量和影院大小。

第二行 n 个正整数 $m_{1 \dots n}$ 依次表示每次要求购买的票数。

【输出格式】

输出到文件 *movie.out* 中。

输出 n 行，每个购买请求的结果为一行。

如果无法在一排中买到 m_i 个连续的座位，则在对应的行中输出 -1 。否则输出三个正整数 x, l, r ，表示所买电影票的排号和起止座位号。

【样例 1 输入】

```
6 3
1 2 3 1 2 1
```

【样例 1 输出】

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

【样例 2】

见选手目录下的 *movie/movie2.in* 与 *movie/movie2.ans*。

【数据范围与提示】

对于所有测试点： $1 \leq n \leq 4 \times 10^5$ ， $1 \leq m_i \leq k \leq 300001$ ， k 为奇数。
每个测试点的具体限制见下表：

测试点编号	$n \leq$	$k \leq$
1 ~ 2	50	25
3 ~ 4	200	101
5	1000	501
6	2000	1001
7	100000	50001
8	200000	100001
9	300000	200001
10	400000	300001