

## 正睿 OI 省队选拔 2020 模拟赛

diamond\_duke

题目名称	游戏	石子	划分
可执行文件名	game	stone	divide
输入文件名	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
时间限制	1s	2s	1s
内存限制	512MB	512MB	512MB
子任务个数	5	10	6
题目类型	传统型	传统型	传统型

**请注意：** 评测时开启 O2 优化和 C++11 编译选项，栈空间限制同空间限制。

# 1 游戏

## 1.1 Problem Statement

小 D 正在玩一个有趣的游戏。

在这个游戏中，小 D 会遇到  $n$  个敌人。这些敌人会依次站成一排，其中第  $i$  个敌人需要 D 花费  $a_i$  的魔法值打死。

除了正常的攻击之外，小 D 还有一个只能发动一次的大招。这个大招花费  $w_b$  的魔法值，可以打死  $k$  名相邻的敌人。

为了更好地使用大招，小 D 还可以花费  $w_x$  的魔法值交换相邻两个敌人的位置。交换技能可以使用任意多次。

小 D 想要知道，打死所有敌人所需要的最少魔法值。

但他并不会，请你帮帮他。

## 1.2 Input Format

从标准输入读入数据。

第一行两个整数  $n, k$ ，表示敌人的个数，以及大招可以打死的敌人个数。

第二行  $n$  个整数  $a_1, a_2, \dots, a_n$ ，表示正常情况下，打死每个敌人需要的魔法值。

第三行两个整数  $w_s, w_b$ ，表示小 D 发动交换技能和大招需要的魔法值。

## 1.3 Output Format

向标准输出输出答案。

输出一行一个整数，表示最少需要的总魔法值。

## 1.4 Sample 1

### 1.4.1 Input

```
5 2
9 1 7 3 6
2 1
```

### 1.4.2 Output

```
13
```

### 1.4.3 Explanation

可以先花费 2 的魔法值交换第一个和第二个数字，然后花费 1 的魔法值使用大招打死现在的第 2,3 个敌人，再使用  $1+3+6$  的魔法值打死剩余的敌人，共花费 13 点魔法值。

### 1.5 Sample 2

见下发文件 `game/game2.in` 与 `game/game2.ans`。

### 1.6 Sample 3

见下发文件 `game/game3.in` 与 `game/game3.ans`。

### 1.7 Constraints

对于所有测试数据， $1 \leq k \leq n \leq 10^6$ ， $1 \leq a_i, w_s, w_b \leq 10^9$ 。

- 子任务 1 (10 分):  $n \leq 10$ ;
- 子任务 2 (20 分):  $n \leq 300$ ;
- 子任务 3 (20 分):  $n \leq 3 \times 10^4$ ;
- 子任务 4 (20 分):  $n \leq 3 \times 10^5$ ;
- 子任务 5 (30 分): 无特殊限制。

## 2 石子

### 2.1 Problem Statement

小 D 和小 Y 正在玩石子游戏。

小 D 和小 Y 画了  $n$  个相邻的格子，其中第  $i$  个格子上写着数字  $a_i$ 。

小 D 和小 Y 在格子中放了  $k$  个石子，且没有两个石子位于同一个格子中。

然后小 D 和小 Y 开始了游戏。他们轮流操作，从小 D 开始，每次操作者可以选择两个相邻的石子，然后将它们删去，再放一个石子在它们之间。

形式化地，当前操作者选择  $l < r$ ，满足第  $l, r$  个格子中有石子，而任意  $x \in (l, r)$  格子中都没有石子。然后，操作者删去位于  $l, r$  的石子，选择一个  $x \in [l, r]$ ，并放一个石子在  $x$  上。

当仅剩一个石子时，游戏停止。此时该石子所在的格子中写着的数字即为游戏结果。小 D 想要最大化该结果，而小 Y 想要最小化它。

小 D 和小 Y 当然知道双方都绝顶聪明时，结果是多少。为了更加有趣，他们会对游戏的初始条件（石子的位置以及格子上的数字）进行一些更改后，再进行同样的游戏。他们同样想要知道每次这样的调整后，结果会是多少。

这下他们不会算了，请你帮帮他们。为了更好地进行之后的调整，你可能需要实时回答这一结果。换言之，部分测试点强制在线。

### 2.2 Input Format

从标准输入读入数据。

第一行四个整数  $n, k, q, T$ ，表示格子的长度，初始时石子的个数，调整的次数，以及是否强制在线。

第二行  $n$  个空格隔开的整数  $a_1, a_2, \dots, a_n$ ，表示初始时每个格子上写的数字。

第三行  $k$  个空格隔开的整数  $p_1, p_2, \dots, p_k$ ，表示初始时有棋子的格子。

接下来  $q$  行，每行 2 或 3 个整数，格式如下：

- 1 x: 表示翻转第  $x$  个格子是否有石子的状态。即，从有石子变为没有，或从没有石子变为有；
- 2 x v: 表示将第  $x$  个格子上写的数字改为  $v$ 。

若  $T = 1$ ，则上述输入中的  $x$  均是经过加密的：若输入的值为  $x'$ ，则真正的  $x$  应当通过  $(x' - 1 + \text{ans}) \bmod n + 1$  求得，其中  $\text{ans}$  为此前输出的最后一个数。

### 2.3 Output Format

向标准输出输出答案。

输出  $q+1$  行，每行一个整数，其中第  $i$  个整数表示第  $i-1$  次调整后的结果。特别地，对于  $i=1$ ，这表示初始情况下进行游戏得到的结果。

## 2.4 Sample 1

### 2.4.1 Input

```
5 2 3 0
1 2 3 4 5
2 4
1 1
1 5
2 3 6
```

### 2.4.2 Output

```
4
2
4
6
```

## 2.5 Sample 2

见下发文件 `stone/stone2.in` 与 `stone/stone2.ans`。

## 2.6 Sample 3

见下发文件 `stone/stone3.in` 与 `stone/stone3.ans`。

## 2.7 Constraints

对于所有测试数据， $1 \leq n, k, q \leq 10^5$ ， $T \in \{0, 1\}$ ， $1 \leq a_i, v \leq 10^9$ ， $1 \leq x, p_i \leq n$ ， $p_i$  中的元素互不相同。

- 子任务 1,2 (7 分, 8 分):  $n \leq 50$ ,  $q = 0$ ;
- 子任务 3,4 (7 分, 8 分):  $n \leq 500$ ,  $q = 0$ ;
- 子任务 5,6 (7 分, 8 分):  $n \leq 5000$ ,  $q = 0$ ;
- 子任务 7,8 (7 分, 8 分):  $q = 0$ ;
- 子任务 9,10 (15 分, 25 分): 无特殊限制。

对于编号为奇数的子任务，我们保证  $T = 0$ 。

## 3 划分

### 3.1 Problem Statement

小 D 正在研究划分。

小 D 写出了一个数字序列  $a_1, a_2, \dots, a_n$ 。

小 D 想要将这些数字划分为若干组。每一组的**权值**被定义为，这组中所有数字的最大公约数。对于一个划分方案，小 D 定义其**价值**为每一组的权值之和。

对于每个  $1 \leq k \leq n$ ，小 D 要求出：所有划分成  $k$  组的方案中，价值的最大值。

形式化地，对于一个集合  $S$ ，我们定义其一个**划分**为集合  $\{T_i\} = T_1, T_2, \dots, T_k$ ，满足：

- 对于任意  $1 \leq i < j \leq k$ ，有  $T_i \cap T_j = \emptyset$ ；
- $\bigcup_{i=1}^n T_i = T_1 \cup T_2 \cup \dots \cup T_n = S$ 。

我们称这样的一个划分**大小**为  $k$ ，记作  $|\{T_i\}| = k$ 。

记  $[n] = \{1, 2, \dots, n\}$ 。对于集合  $S \subseteq [n]$ ，我们定义其权值  $W(S) = \gcd(a_i \mid i \in S)$ 。则对于集合  $[n]$  的一个划分  $\{S_i\}$ ，我们定义其权值为  $V(\{S_i\}) = \sum_{i=1}^{|\{S_i\}|} W(S_i)$ 。小 D 即想要对于每个  $k$ ，求出  $\max_{|\{S_i\}|=k} V(\{S_i\})$  的值，其中  $\{S_i\}$  是  $[n]$  的一个划分。

但他并不会，请你帮帮他。

### 3.2 Input Format

从标准输入读入数据。

第一行一个整数  $n$ ，表示小 D 写出的序列长度。

第二行一行  $n$  个整数  $a_1, a_2, \dots, a_n$ ，表示小 D 写出的序列。

### 3.3 Output Format

向标准输出输出答案。

输出  $n$  行，每行一个整数，其中第  $i$  行的整数表示  $k = i$  时的最大权值。

### 3.4 Sample 1

#### 3.4.1 Input

```
3
2 4 9
```

### 3.4.2 Output

1  
11  
15

### 3.4.3 Explanation

- 当  $k = 1$  时, 所有数字都必须划分在一组, 因此权值为  $\gcd(2, 4, 9) = 1$ ;
- 当  $k = 2$  时, 最优方案是将 2, 4 划分在一组, 9 划分在另一组, 因此权值为  $\gcd(2, 4) + \gcd(9) = 11$ ;
- 当  $k = 3$  时, 最优方案是每个数字各自一组, 因此权值为  $\gcd(2) + \gcd(4) + \gcd(9) = 15$ 。

### 3.5 Sample 2

见下发文件 `divide/divide2.in` 与 `divide/divide2.ans`。

### 3.6 Sample 3

见下发文件 `divide/divide3.in` 与 `divide/divide3.ans`。

### 3.7 Constraints

对于所有测试数据,  $1 \leq n \leq 5 \times 10^5$ ,  $1 \leq a_i \leq 10^{12}$ 。

- 子任务 1 (10 分):  $n \leq 15$ ;
- 子任务 2 (10 分):  $n \leq 100$ ,  $a_i \leq 500$ ;
- 子任务 3 (15 分):  $n \leq 2000$ ,  $a_i \leq 2000$ ,  $a_i$  互不相同;
- 子任务 4 (15 分):  $n \leq 2000$ ,  $a_i \leq 2000$ ;
- 子任务 5 (20 分):  $a_i$  互不相同;
- 子任务 6 (30 分): 无特殊限制。