

# 2023 年全国青少年信息学奥林匹克联赛

## CQBZ NOIP 2023

### 第一试

时间: 2023 年 10 月 27 日 08:00 ~ 12:30

题目名称	菜菜菜	缺金木	农民王	计划
题目类型	传统型	传统型	传统型	传统型
目录	vegetable	kaneki	king	plan
可执行文件名	vegetable	kaneki	king	plan
输入文件名	vegetable.in	kaneki.in	king.in	plan.in
输出文件名	vegetable.out	kaneki.out	king.out	plan.in
每个测试点时限	1.0 秒	1.0 秒	3.0 秒	1.0 秒
内存限制	256 MB	256 MB	256 MB	256 MB
测试点数目	10	4	20	5
测试点是否等分	是	否	是	是

提交源程序文件名

对于 C++ 语言	vegetable.cpp	kaneki.cpp	king.cpp	plan.cpp
-----------	---------------	------------	----------	----------

编译选项

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

#### 注意事项与提醒 (请选手务必仔细阅读)

1. 选手提交的源程序必须存放在已建立好的, 且带有**样例文件**和**下发文件**的文件夹中, 文件夹名称与对应试题英文名一致。
2. 文件名 (包括程序名和输入输出文件名) 必须使用英文小写。
3. C++ 中函数 main() 的返回值类型必须是 int, 值必须为 0。
4. **对于因未遵守以上规则对成绩造成的影响, 相关申诉不予受理。**
5. 若无特殊说明, 结果比较方式为**忽略行末空格、文末回车后的全文比较**。
6. 程序可使用的栈空间大小与该题内存空间限制一致。
7. 在终端中执行命令 `ulimit -s unlimited` 可将当前终端下的栈空间限制放大, 但你使用的栈空间大小不应超过题目限制。
8. 每道题目所提交的**代码文件大小限制为 100KB**。
9. 若无特殊说明, 输入文件与输出文件中同一行的相邻整数均使用一个空格分隔。
10. 输入文件中可能存在行末空格, 请选手使用更完善的读入方式 (例如 `scanf` 函数) 避免出错。

11. 直接复制 PDF 题面中的多行样例，数据将带有行号，建议选手直接使用对应目录下的样例文件进行测试。

12. 使用 `std::deque` 等 STL 容器时，请注意其内存空间消耗。

13. 请务必使用题面中规定的的编译参数，保证你的程序在本机能够通过编译。此外不允许在程序中手动开启其他编译选项，一经发现，本题成绩以 0 分处理。

## 菜菜菜 (vegetable)

### 【题目描述】

Alice 和 Bob 都不喜欢吃蔬菜，而现在他们俩正在超市大促中参与瓜分蔬果的活动。

具体而言，现在有一排蔬果，一共  $n$  个，其中  $V$  代表蔬菜， $F$  代表水果。

由 Alice 先手开始，每次两人轮流选择最左边或者最右边的蔬果，拿走它。

如果 Alice 或者 Bob 中有一个人拿到了  $k$  个蔬菜，他就会破防跑路，而另一个人会得到剩下的所有蔬果。假设他们都非常聪明，请告诉 Alice 对于某个蔬果排列她是否一定能获胜？

### 【输入格式】

从文件 `vegetable.in` 中读入数据。

第一行一个正整数  $t$ ，表示测试组数，接下来  $t$  组数据，每组数据格式为：

第一行两个正整数  $n, k$ ，分别为蔬果串的长度和两个人破防需要拿到的蔬果数  $k$ 。

接下来一行给出一个长度为  $n$  的字符串，表示蔬果串。

数据保证蔬菜的数量不少于  $2k-1$ 。

### 【输出格式】

输出到文件 `vegetable.out` 中。

输出  $t$  行，每行一个字符串，如果 Alice 一定获胜就输出 `yes`，否则输出 `no`。

### 【样例 1 输入】

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

### 【样例 1 输出】

```
1 yes
```

### 【样例 1 解释】

只要 Alice 拿走最右端的水果，Bob 怎么样都会拿到蔬菜并且破防，所以 Alice 必胜。

### 【样例 2】

见下发文件。

**【数据范围】**

本题为多测，请注意清空。

对于 20% 的数据保证， $k \leq n \leq 20$ 。

对于另外 20% 的数据保证， $k \leq n \leq 50$ 。

对于 100% 的数据保证， $k \leq n \leq 350, t \leq 5$ 。

## 缺金木 (kaneki)

### 【题目描述】

CLJ 一共有  $n$  个新节目，每个节目的惊险值为  $a_i$ 。

接下来 CLJ 可以通过若干次操作来对每个节目的惊险值进行变换：

- CLJ 使用冰球，对于所有的  $l \leq i \leq r$ ，第  $i$  个节目的惊险值会增加  $x$ 。
- CLJ 使用土球，对于所有的  $l \leq i \leq r$ ，第  $i$  个节目的惊险值会乘以  $x$ 。
- CLJ 使用火球，对于所有的  $l \leq i \leq r$ ，第  $i$  个节目的惊险值在接下来的  $x$  个操作中不会受到冰球和土球的影响。火球效果不会被替换。

当然观众对每个节目的惊险值也很好奇，所以你需要在操作过程中帮 CLJ 回答对于所有  $l \leq i \leq r$ ，第  $i$  个节目的惊险值之和。当然观众不希望惊险值过大，所以你需要将结果对  $10^9 + 7$  取模。

**简要题意：**给你一个长度为  $n$  的数组  $a$ ，你需要支持如下几个操作：

1. 'l r x'：对所有  $l \leq i \leq r$ ，令  $a_i$  增加  $x$ 。
2. 'l r x'：对所有  $l \leq i \leq r$ ，令  $a_i$  乘上  $x$ 。
3. 'l r x'：对所有  $l \leq i \leq r$ ，在接下来的  $x$  个操作内， $a_i$  将会被封锁，不会受到操作 1 和操作 2 的影响（设本次操作为第  $k$  次操作，则第  $k+1, k+2, \dots, k+x$  次操作中的所有操作 1 和操作 2 不会对区间  $[l, r]$  产生影响）。封锁效果不会被替换，即如果第 3 次有一个操作 3 对某个位置进行封锁，封锁时间为 5，第 5 次操作同样对这个位置进行封锁，封锁时间为 2 那么实际上这个位置在第 4 次操作到第 8 次操作的时间内都会被封锁（感性理解就是后面时间短的封锁不会使前面时间长的封锁失效）。
4. 'l r'：询问  $\sum_{l \leq i \leq r} a_i$ ，对  $10^9 + 7$  取模。

### 【输入格式】

从文件 *kaneki.in* 中读入数据。

第一行有两个正整数  $n, m$ ，分别表示数组长度和操作数。

第二行会有  $n$  个整数  $a_i$ ，表示数组  $a$ 。

接下来  $m$  行，每行有三个整数  $opt, l, r$  表示操作编号以及这次操作的  $l, r$ ，如果  $opt \leq 3$ ，那么还会有一个整数  $x$ ，表示这次操作的  $x$ 。

### 【输出格式】

输出到文件 *kaneki.out* 中。

对于所有的操作 4，一行一个整数表示结果，对  $10^9 + 7$  取模。

**【样例 1 输入】**

```
1 5 5
2 1 5 4 3 6
3 1 2 4 3
4 3 1 2 2
5 4 2 5
6 2 2 3 4
7 4 1 3
```

**【样例 1 输出】**

```
1 27
2 37
```

**【样例 1 解释】**

一开始数组为  $\{1, 5, 4, 3, 6\}$ 。

- 执行第 1 次操作，此时数组变为  $\{1, 8, 7, 6, 6\}$ 。
- 执行第 2 次操作，此时数组不变。
- 执行第 3 次操作，询问结果为 27。
- 执行第 4 次操作，因为此时  $a_2$  在第 2 次操作中被封锁还未解除，所以这次操作只对  $a_3$  产生影响，数组变为  $\{1, 8, 28, 6, 6\}$ 。
- 执行第 5 次操作，询问结果为 37。

**【样例 2 输入】**

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

```
9 2 4 5 2
10 3 6 8 2
11 4 2 3
12 1 2 10 6
13 2 7 9 3
14 4 1 10
```

**【样例 2 输出】**

```
1 129
2 16
3 314
```

**【数据范围】**

本题采用捆绑测试。

对于全部数据,  $1 \leq n, m \leq 2 \times 10^5, 0 \leq a_i < 10^9 + 7, 1 \leq l \leq r \leq n$ , 对于所有操作 1 和操作 2, 保证  $0 \leq x < 10^9 + 7$ , 对于所有操作 3, 设其为第  $k$  次操作, 保证  $0 \leq x \leq m - k$ 。

子任务 1 ( 25% ):  $n, m \leq 2 \times 10^3$ 。

子任务 2 ( 8% ): 没有操作 3。

子任务 3 ( 17% ): 对于所有操作 4, 保证  $l = r$ 。

子任务 4 ( 50% ): 无特殊限制。

## 农民王 (king)

### 【题目描述】

有这么一个人叫做农民王 Q，他有一个家谱，现在他在想自己和上古农民王 U 到底有多大的关系，关系式中有这么些个亲戚：

“father, mother, son, daughter, husband, wife, brother, sister, grandfather, grandmother, grandson, granddaughter, uncle, aunt, nephew, niece”

中文意思分别是：“父亲，母亲，儿子，女儿，丈夫，妻子，兄弟，姐妹，爷爷，奶奶，孙子，孙女，叔叔，阿姨，侄儿，侄女”

对于亲戚关系，满足以下几点：

1. Q 的兄弟等同于 Q 的父亲的或者母亲的儿子 (Q 自己除外)；
2. Q 的爷爷等同于 Q 的父亲的或者母亲的父亲；
3. Q 的孙子等同于 Q 的儿子的或者女儿的儿子；
4. Q 的叔叔等同于 Q 的父亲的或者母亲的兄弟；
5. Q 的侄儿等同于 Q 的兄弟的或者姐妹的儿子；
6. 上述规则对于姐妹，奶奶，孙女，阿姨和侄女类似。

血缘关系的定义如下：

1. Q 到 Q 的父亲，Q 的母亲，Q 的儿子或者 Q 的女儿的距离为 1；
2. Q 到 Q 的丈夫 (妻子) 的距离为 0；
3. Q 到 U 的距离等于在上述规则下推断出的 Q 到 U 的最短距离。

由于一条关系会出现很多种情况，所以农民王想知道他跟上古农民王的血缘关系距离究竟有多少种？分别是多少？请你帮帮他好吗？

注明：不会出现的关系包括收养，亲戚间结婚 (家族树中无环)，离婚，复婚，重婚，同性恋等。

### 【输入格式】

从文件 *king.in* 中读入数据。

第 1 行包括一个字符串表示氏族谱图上的关系式，格式如下：

Q is U' s relation' s relation' s ... relation

设关系式中出现的亲戚关系总个数为  $l$ 。

### 【输出格式】

输出到文件 *king.out* 中。



第 1 行一个整数  $c$ ，表示一共有多少种情况。

第 2 行  $c$  个数，表示每种情况的距离，空格隔开，按升序输出。

**【样例 1 输入】**

```
1 Q is U's father's brother's son's aunt
```

**【样例 1 输出】**

```
1 2
2 3 5
```

**【样例 2 输入】**

```
1 Q is U's mother's brother's son's aunt
```

**【样例 2 输出】**

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

**【样例 3 输入】**

```
1 Q is U's son's mother's mother's son
```

**【样例 3 输出】**

```
1 1
2 2
```

**【样例 4 ~ 7】**

见下发文件。

**【数据范围】**

对于 10% 的数据， $l \leq 10$ 。

对于 30% 的数据， $l \leq 30$ 。

对于另外 20% 的数据，只包含“父亲，母亲，儿子，女儿，丈夫，妻子，兄弟，姐妹”关系，且兄弟姐妹关系个数不超过 20 个

对于全部数据， $0 \leq l \leq 100$ ，不保证一定存在一种情况满足关系式。

## 计划 (plan)

### 【题目描述】

你在一家商店内购物，总共有  $N$  件商品。其中第  $i$  件商品的类型是  $a_i$ ，它是一个在  $1 \sim M$  之间的整数。

一个可行的购物计划（也就是选取这些商品的一个子集）必须满足：对于类型为  $j$  的所有商品，被选中的个数必须在  $[x_j, y_j]$  之间。

第  $i$  件商品的价格为  $c_i$ ，而购物计划的代价就是所有选中的商品的价格之和。

请你求出最小的  $K$  个可行的购物计划的代价。**注意如果两个不同的购物计划有着相同的代价，你也应该要分别输出。**

### 【输入格式】

从文件 *plan.in* 中读入数据。

第一行三个正整数  $N, M, K$ 。

接下来  $N$  行，第  $i$  行两个正整数  $a_i, c_i$ 。

接下来  $M$  行，第  $j$  行两个整数  $x_j, y_j$ 。

### 【输出格式】

输出到文件 *plan.out* 中。

输出  $K$  行，第  $i$  行输出第  $i$  小的计划的代价，如果可行的计划数量不足  $i$  个，则输出  $-1$ 。

### 【样例 1 输入】

```
1 5 2 7
2 1 5
3 1 3
4 2 3
5 1 6
6 2 1
7 1 1
8 1 1
```

### 【样例 1 输出】

```
1 4
2 6
3 6
4 7
```

5	8
6	9
7	-1

**【样例 1 解释】**

一个可行的购物计划必须包含恰好一个价格属于  $\{5, 3, 6\}$  的商品和恰好一个价格属于  $\{3, 1\}$  的商品。

**【样例 2 ~ 3】**

见下发文件。

**【数据范围】**

本题采用捆绑测试。

对于全部数据,  $1 \leq N, M, K \leq 2 \times 10^5, 1 \leq a_i \leq M, 1 \leq c_i \leq 10^9, 0 \leq x_j \leq y_j \leq N$ 。

子任务 1 :  $x_j = y_j = 1$  并且  $N, M, K \leq 4000$ 。

子任务 2 :  $x_j = y_j = 1$  并且  $N, M, c_i \leq 4000$ 。

子任务 3 :  $x_j = y_j = 1$ 。

子任务 4 :  $x_j = 0$ 。

子任务 5 : 无特殊限制。