

# 全国青少年信息学奥林匹克竞赛

## CCF NOI2020 统一省选

Day1

时间：2020 年 6 月 20 日 08:30 ~ 13:00

题目名称	冰火战士	组合数问题	魔法商店
题目类型	传统型	传统型	传统型
目录	icefire	problem	shop
可执行文件名	icefire	problem	shop
输入文件名	icefire.in	problem.in	shop.in
输出文件名	icefire.out	problem.out	shop.out
每个测试点时限	3.0 秒	1.0 秒	1.0 秒
内存限制	512 MB	512 MB	512 MB
子任务数目	10	20	20
测试点是否等分	是	是	是

提交源程序文件名

对于 C++ 语言	icefire.cpp	problem.cpp	shop.cpp
-----------	-------------	-------------	----------

编译选项

对于 C++ 语言	-lm	-O2	-O2
-----------	-----	-----	-----

注意事项与提醒（请选手务必仔细阅读）

1. 文件名（程序名和输入输出文件名）必须使用英文小写。
2. 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是 0。
3. 提交的程序代码文件的放置位置请参照各省的具体要求。
4. 因违反以上三点而出现的错误或问题，申诉时一律不予受理。
5. 若无特殊说明，结果的比较方式为全文比较（过滤行末空格及文末回车）。
6. 程序可使用的栈内存空间限制与题目的内存限制一致。
7. 全国统一评测时采用的机器配置为: Intel(R) Core(TM) i7-8700K CPU @ 3.70GHz, 内存 32GB。上述时限以此配置为准。
8. 评测在当前最新公布的 NOI Linux 下进行，各语言的编译器版本以其为准。
9. 最终评测时所用的编译命令中不含任何优化开关。
10. 最终评测时，程序运行时间取其用户 CPU 时间（user time）。

## 冰火战士 (icefire)

### 【题目描述】

一场比赛即将开始。

每位战士有两个属性：温度和能量，有两派战士：冰系战士的技能会对周围造成降温冰冻伤害，因而要求场地温度不低于他的自身温度才能参赛；火系战士的技能会对周围造成升温灼烧伤害，因而要求场地温度不高于他的自身温度才能参赛。

当场地温度确定时，双方能够参赛的战士分别排成一队。冰系战士按自身温度从低到高排序，火系战士按自身温度从高到低排序，温度相同时能量大的战士排在前面。首先，双方的第一位战士之间展开战斗，两位战士消耗相同的能量，能量少的战士将耗尽能量退出比赛，而能量有剩余的战士将继续和对方的下一位战士战斗（能量都耗尽则双方下一位战士之间展开战斗）。如此循环，直至某方战士队列为空，比赛结束。

你需要寻找最佳场地温度：使冰火双方消耗总能量最高的温度的最高值。

现在，比赛还处于报名阶段，目前还没有任何战士报名，接下来你将不断地收到报名信息 and 撤回信息。其中，报名信息包含报名战士的派系和两个属性，撤回信息包含要撤回的报名信息的序号。每当报名情况发生变化（即收到一条信息）时，你需要立即报出当前局面下的最佳场地温度，以及该场地温度下双方消耗的总能量之和是多少。若当前局面下无论何种温度都无法开展比赛（某一方没有战士能参赛），则只要输出 **Peace**。

### 【输入格式】

从文件 *icefire.in* 中读入数据。

第一行一个数  $Q$ ，表示信息的数量。

接下来  $Q$  行，每行为  $1\ t\ x\ y$  ( $t \in \{0, 1\}$ ,  $x$  和  $y$  都是正整数) 或  $2\ k$  ( $k$  是正整数)：

$1\ t\ x\ y$  表示一条报名信息， $t = 0$  时报名战士是冰系， $t = 1$  时报名战士是火系， $x$  表示战士的自身温度， $y$  表示战士的能量。

$2\ k$  表示一条撤回信息，撤回的是第  $k$  条信息。被撤回的信息一定是报名信息，已被撤回的信息不会再次被撤回。

### 【输出格式】

输出到文件 *icefire.out* 中。

共  $Q$  行，每行有两个用空格隔开的正整数，分别表示当前局面下的最佳温度和该温度下冰火双方消耗的总能量之和。

### 【样例 1 输入】

```
1 8
2 1 1 103 150
3 1 0 100 100
4 1 1 102 150
5 1 0 103 300
6 2 1
7 1 1 101 100
8 1 1 104 350
9 1 0 100 400
```

**【样例 1 输出】**

```
1 Peace
2 103 200
3 103 200
4 103 300
5 102 200
6 102 200
7 104 700
8 102 1000
```

**【样例 1 解释】**

为说明方便，约定：若第  $k$  条信息是报名信息，则该条报名信息对应战士  $k$ 。样例中含有战士 1、2、3、4、6、7、8，由于第 5 条是撤回信息，所以没有战士 5。

下面逐个解释每个输出：

1. 只有火系战士：战士 1，无法比赛，输出 Peace。
2. 温度为 100 ~ 103 都能消耗最多的能量 200：战士 1 对阵战士 2 消耗能量 200，最佳温度为 103。
3. 温度为 100 ~ 103 都能消耗最多的能量 200：战士 1 对阵战士 2 消耗能量 200，最佳温度为 103。
4. 温度 103 能消耗最多的能量 300：首先，战士 1 对阵战士 2 消耗能量 200；然后，战士 1 对阵战士 4 消耗能量 100，最佳温度为 103。
5. 从现在起战士 1 不再存在。温度 100 ~ 102 能消耗最多的能量 200：战士 2 对阵战士 3 消耗能量 200，最佳温度为 102。

6. 温度  $100 \sim 102$  能消耗最多的能量 200: 战士 2 对阵战士 3 消耗能量 200, 最佳温度为 102。
7. 温度  $100 \sim 104$  能消耗最多的能量 700: 首先, 战士 2 对阵战士 7 消耗能量 200; 然后, 战士 4 对阵战士 7 消耗能量 500, 最佳温度为 104。
8. 温度  $100 \sim 102$  能消耗最多的能量 1000: 首先, 战士 7 对阵战士 8 消耗能量 700; 然后, 战士 1 对阵战士 8 消耗能量 100; 接着, 战士 1 对阵战士 2 消耗能量 200, 最佳温度为 102。

### 【样例 2】

见选手目录下的 *icefire/icefire2.in* 与 *icefire/icefire2.ans*。

### 【数据范围】

10% 的数据:  $Q \leq 100, x \leq 10^3$ 。

另有 20% 的数据:  $Q \leq 10^4, x \leq 5000$ , 不存在撤回信息, 且输入的  $x$  按顺序不降。

60% 的数据 (包含上述 20%, 下同):  $Q \leq 2 \times 10^5, x \leq 2 \times 10^5$ 。

90% 的数据:  $Q \leq 2 \times 10^6, x \leq 2 \times 10^6$ 。

100% 的数据:  $1 \leq Q \leq 2 \times 10^6, 1 \leq x \leq 2 \times 10^9$ , 所有  $y$  之和不超过  $2 \times 10^9$ , 保证不存在  $t, x, y$  完全相同的两个战士。

## 组合数问题 (problem)

### 【题目描述】

众所周知，小葱同学擅长计算，尤其擅长计算组合数。小葱现在希望你计算

$$\left( \sum_{k=0}^n f(k) \times x^k \times \binom{n}{k} \right) \bmod p$$

的值。其中  $n, x, p$  为给定的整数， $f(k)$  为给定的一个  $m$  次多项式  $f(k) = a_0 + a_1k + a_2k^2 + \cdots + a_mk^m$ 。  $\binom{n}{k}$  为组合数，其值为  $\binom{n}{k} = \frac{n!}{k!(n-k)!}$ 。

### 【输入格式】

从文件 `problem.in` 中读入数据。

第一行四个非负整数  $n, x, p, m$ 。

第二行  $m+1$  个整数，分别代表  $a_0, a_1, \cdots, a_m$ 。

### 【输出格式】

输出到文件 `problem.out` 中。

仅一行一个整数表示答案。

### 【样例 1 输入】

```
1 5 1 10007 2
2 0 0 1
```

### 【样例 1 输出】

```
1 240
```

### 【样例 1 解释】

$f(0) = 0, f(1) = 1, f(2) = 4, f(3) = 9, f(4) = 16, f(5) = 25$ 。

$x = 1$ ，故  $x^k$  恒为 1，乘积中的该项可以忽略。

$\binom{5}{0} = 1, \binom{5}{1} = 5, \binom{5}{2} = 10, \binom{5}{3} = 10, \binom{5}{4} = 5, \binom{5}{5} = 1$ 。

最终答案为：

$$\sum_{k=0}^5 f(k) \times \binom{5}{k} = 0 \times 1 + 1 \times 5 + 4 \times 10 + 9 \times 10 + 16 \times 5 + 25 \times 1 = 240$$

## 【样例 2 输入】

```
1 996 233 998244353 5
2 5 4 13 16 20 15
```

## 【样例 2 输出】

```
1 869469289
```

## 【样例 3】

见选手目录下的 *problem/problem3.in* 与 *problem/problem3.ans*。

## 【数据范围与提示】

对于所有测试数据： $1 \leq n, x, p \leq 10^9$ ， $0 \leq a_i \leq 10^9$ ， $0 \leq m \leq \min(n, 1000)$ 。  
每个测试点的具体限制见下表：

测试点编号	$n \leq$	$m \leq$	其他特殊限制
1 ~ 3	1000	1000	无
4 ~ 6	$10^5$	0	$p$ 是质数
7 ~ 8	$10^9$		无
9 ~ 12		5	
13 ~ 16		1000	$x = 1$
17 ~ 20			无

## 魔法商店 (shop)

### 【题目描述】

笠笠和伦伦来到了一家魔法商店，这家商店内有  $n$  件礼品，礼品从  $1 \sim n$  编号， $i$  号礼品的魅力值为  $c_i$ ，价格为  $v_i$ 。

两人希望购买一些礼品，但他们的要求比较奇怪：假设购买到的礼品集合为  $S = \{s_1, s_2, \dots, s_p\}$  ( $1 \leq s_i \leq n$ )，两人要求对于  $S$  中任意的非空子集  $T = \{t_1, t_2, \dots, t_q\}$ ，它包含的所有礼品的魅力值异或和都不为零，即： $c_{t_1} \oplus c_{t_2} \oplus \dots \oplus c_{t_q} \neq 0$ 。其中  $\oplus$  是异或运算。在此基础上，两人还要求购买到的礼品数尽可能多。

例如： $c_1 = 1, c_2 = 2, c_3 = 5, c_4 = 6, c_5 = 7$ 。则  $S_1 = \{2, 3, 5\}$  不符合要求，因为  $c_2 \oplus c_3 \oplus c_5 = 0$ 。 $S_2 = \{1, 2, 3\}$  与  $S_3 = \{2, 4, 5\}$  符合要求，其任意非空子集的异或和都不为零。 $S_4 = \{1, 2\}$  不符合要求，因为其包含的礼品数不是最多的。

满足两人要求的礼品集合可能很多，因此商店老板为两人挑选出了两个符合要求的礼品集合  $A$  与  $B$ （显然它们所含的礼品数相同），伦伦喜欢集合  $A$ ，但笠笠更喜欢集合  $B$ 。为了笠笠同意购买集合  $A$ ，伦伦决定使用魔法改变礼品价格。更具体地，伦伦能花费  $(x - v_i)^2$  的魔力值，将  $i$  号礼品的价格改为任意整数  $x$ ，每件礼品只能被改价一次。

伦伦希望改价后  $A$  是所有符合要求的礼品集合之中价格总和最小的，且  $B$  是所有符合要求的礼品集合之中价格总和最大的（一个礼品集合的价格总和为它包含的所有礼品的价格之和）。现在请你帮伦伦计算，他至少要花费多少魔力值才能完成他的目标。

### 【输入格式】

从文件 **shop.in** 中读入数据。

第一行两个整数  $n, m$ ，分别表示总礼品数与礼品集合  $A$  ( $B$ ) 包含的礼品数。

第二行  $n$  个整数  $c_i$ ，第  $i$  个整数表示  $i$  号礼品的魅力值。

第三行  $n$  个整数  $v_i$ ，第  $i$  个整数表示  $i$  号礼品的价格。

第四行  $m$  个整数  $a_i$ ，表示礼品集合  $A$  包含的礼品的编号。数据保证  $a_i$  两两不同。

第五行  $m$  个整数  $b_i$ ，表示礼品集合  $B$  包含的礼品的编号。数据保证  $b_i$  两两不同。数据保证  $1 \leq a_i, b_i \leq n$ ，且礼品集合  $A$  和  $B$  均符合两人的要求。

### 【输出格式】

输出到文件 **shop.out** 中。

仅一行一个整数，表示伦伦至少需要花费的魔力值。

### 【样例 1 输入】



```

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

### 【样例 1 输出】

```

1 6
    
```

### 【样例 1 解释】

符合条件的礼品集合有：{1, 2, 3}, {1, 2, 4}, {1, 2, 5}, {1, 3, 4}, {1, 3, 5}, {2, 3, 4}, {2, 4, 5}, {3, 4, 5}。

一个最优的改价方案为： $c_1 = c_2 = c_4 = c_5 = 3$ ,  $c_3 = 2$ 。

### 【样例 2】

见选手目录下的 *shop/shop2.in* 与 *shop/shop2.ans*。

### 【样例 3】

见选手目录下的 *shop/shop3.in* 与 *shop/shop3.ans*。

### 【数据范围与提示】

对于所有测试数据： $1 \leq n \leq 1000$ ,  $1 \leq m \leq 64$ ,  $1 \leq c_i < 2^{64}$ ,  $0 \leq v_i \leq 10^6$ 。

每个测试点的具体限制见下表：

测试点编号	$n \leq$	$m \leq$	特殊限制
1 ~ 3	10	4	$1 \leq v_i \leq 5$
4 ~ 6	50	2	$1 \leq v_i \leq 10$
7 ~ 10	500	30	$0 \leq v_i \leq 1$
11 ~ 12	1000	64	$A$ 与 $B$ 相同
13 ~ 20			无