

# 模拟赛

## 【题目信息】

序号	A	B	C	D
题目中文名	糖果	汉堡	酒吧	三明治
题目英文名	candy	hamburger	bar	sandwich
输入文件	candy.in	hamburger.in	bar.in	sandwich.in
输出文件	candy.out	hamburger.out	bar.out	sandwich.out
时间限制	2s	1s	2s	8s
空间限制	512MB	512MB	512MB	512MB
题目类型	传统题	传统题	传统题	传统题
测试点数量	20	20	20	---

## 【提交源程序文件名】

	A	B	C	D
对于 C++ 语言	candy.cpp	hamburger.cpp	bar.cpp	sandwich.cpp

注意事项：

1. 编译选项：`-lm -O2 -std=c++14 -w1,--stack=998244353`。
2. C++ 中的 `main()` 函数返回值类型必须是 `int`，程序正常结束时的返回值必须是 0。
3. 所有题目的答案比较方式为忽略行末空格和文末回车后的全文比较。
4. **前两题按测试点得分；第三题对于前 50% 的数据按点得分，对于后 50% 的数据捆绑测试且依赖于前 50% 的数据；第四题按子任务得分，且启用子任务依赖。**
5. **题目大致按难度顺序排列，但请选手自行判断。**
6. 部分题目读入量较大，请使用较快的读入方式。
7. 选手应将各题的源程序放在选手文件夹内，**不要建立子文件夹**。
8. 评测机系统为 Windows，64 位，使用 LemonLime 评测。
9. 评测机配置：Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz 1.80 GHz，内存 8G。
10. **由于评测机速度较慢，请注意常数对程序运行时间的影响，但保证所有题目所有测试点的时空限制处于 std 同等环境下运行所需的两倍以上。**

# 糖果（candy）

河边独自看星宿，夜织天丝难接续。——王建《七夕曲》

## 【题目描述】

桌上有  $N$  颗糖，从第 1 颗到第  $N$  颗依次排开，第  $i$  颗糖有甜度  $A_i$ ，保证  $A_i$  是一个  $1, 2, \dots, N$  的排列，你需要进行如下操作，将  $A_i$  变的单调递增，即让所有  $A_i = i$ 。

操作：在  $N$  颗糖中选择任意个，从小到大记录选中的糖的下标为  $x_1, x_2, \dots, x_k$ ，从小到大记录未选中的糖的下标为  $y_1, y_2, \dots, y_{n-k}$ ，然后将所有  $x$  放到所有  $y$  的前面，保持  $x$  内部相对顺序不变，保持  $y$  内部相对顺序不变，即，新的甜度序列  $A$  为  $A_{x_1}, \dots, A_{x_k}, A_{y_1}, \dots, A_{y_{n-k}}$ 。

比如，在  $A_1, A_2, \dots, A_7$  中，选中的  $x = \{1, 3, 5\}$ ，则未选中的  $y = \{2, 4, 6, 7\}$ ，新的  $A$  为  $A_1, A_3, A_5, A_2, A_4, A_6, A_7$ 。

容易证明，可以在有限次操作内达到目的。

求出最小操作次数。

## 【输入格式】

第一行一个整数  $T$ ，表示有  $T$  组数据。

接下来  $T$  组数据，每组数据两行。

对于每组数据，第一行一个整数  $N$ ，第二行  $N$  个整数  $A_i$ 。

## 【输出格式】

对于每组数据，输出一行一个整数，表示最小操作次数。

## 【样例】

### 样例 1 输入

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

### 样例 1 输出

```
2
3
```

### 样例 1 解释

对于第一组数据，操作过程如下：

$\{5, 1, 3, 2, 4\} \rightarrow \{1, 2, 5, 3, 4\} \rightarrow \{1, 2, 3, 4, 5\}$ 。

对于第二组数据，操作过程如下：

$\{6, 5, 2, 4, 1, 3\} \rightarrow \{6, 4, 1, 5, 2, 3\} \rightarrow \{6, 1, 2, 3, 4, 5\} \rightarrow \{1, 2, 3, 4, 5, 6\}$ 。

操作过程可能不唯一。

样例 2 输入/输出

见下发文件中的 *candy2.in/ans*，此样例满足测试点 20 的限制。

数据范围与约定

对于所有数据，满足  $T \in [1, 5], N \in [1, 10^5]$ 。

测试点编号	约束
1, 2	$N \leq 8$
3, 4, 5	$N \leq 1000$
6, 7, 8, 9	$N \leq 10^4$
10, 11, 12, 13	$N \leq 3 \times 10^4$
14, 15, 16, 17, 18, 19, 20	无特殊限制

# 汉堡 (hamburger)

## 【题目描述】

有  $P$  种不相同的面饼和  $Q$  种不相同的肉饼，每种可以选择多次。你需要从中选出至多  $N$  块面饼和至多  $N$  块肉饼制作成汉堡，满足如下条件：

- 所有面饼都在肉饼的上方。
- 考虑到部分客户的特殊口味，可以没有面饼或没有肉饼，但不能没有饼。
- 你需要在 0 或 1 或 2 块饼涂上酱料，特别的，对于涂 2 块的情况，为了避免口味单一，不可以将酱料涂在同一种饼上。

你需要求出不同的汉堡个数，对  $M$  取模，两个汉堡不同，当且仅当饼数不同，或存在同一位置对应的饼不同，或存在同一位置涂酱料的情况不同。

## 【输入格式】

第一行一个整数  $T$ ，表示有  $T$  组数据。

接下来  $T$  行，每行四个整数  $P, Q, N, M$ 。

## 【输出格式】

对于每组数据，输出一个整数，表示答案对  $M$  取模后的结果。

## 【样例】

### 样例 1 输入

```
4
1 1 1 998244353
2 2 3 998244353
1 1 1000000000 1000000000
123 456 789 987654321
```

### 样例 1 输出

```
8
2400
0
345494202
```

### 样例 1 解释

对于第一组数据，符合条件的汉堡如下（用  $A$  表示面饼， $B$  表示肉饼，右上角带 \* 的是涂酱料的饼）。

$A, A^*, B, B^*, AB, A^*B, AB^*, A^*B^*$ 。

样例 2 输入/输出

见下发文件中的 *hamburger2.in/ans*, 此样例满足测试点 20 的限制, 且对于前 5 组测试数据, 有  $P, Q, N \leq 10^3$ ; 对于 6 至 10 组数据, 有  $P = 0$ ; 对于 11 至 15 组数据,  $M$  为质数。

数据范围与约定

对于所有数据, 满足  $T \in [1, 10^3], P, Q, N \in [1, 10^9], M \in [10^8, 10^9]$ 。

测试点编号	约束
1, 2	$T \leq 10, P, Q, N \leq 10$
3, 4	$P, Q, N \leq 10^3$
5, 6, 7, 8	$P = 0$
9, 10	$M$ 为质数
11, 12, 13, 14, 15, 16, 17, 18, 19, 20	无特殊限制

# 酒吧（bar）

举杯邀明月，对影成三人。——李白《月下独酌》

## 【题目描述】

在一条街上有  $N$  家酒吧，每一家酒吧有  $a_i$  种酒提供。

不幸的是，小 Z 已经忘记了  $a_i$  的具体值，只记得其满足三个条件。

$$a_i \geq 0$$

$$\text{xor}_{i=1}^N a_i = X$$

$$\sum_{i=1}^N a_i = S$$

小 Z 想知道，在所有满足他记忆的方案中， $\max_{i=1}^N a_i$  的最小可能值是多少？

## 【输入格式】

第一行一个整数  $T$ ，表示有  $T$  组数据。

接下来  $T$  行，每行三个整数  $N, S, X$ 。

## 【输出格式】

对于每一组测试数据，输出一行一个整数，即  $\max_{i=1}^N a_i$  的最小可能值，特别的，如果不存在合法的  $a_i$ ，输出  $-1$ 。

## 【样例】

### 样例 1 输入

```
6
3 9 3
4 8 0
6 19 1
1 15 15
2 6 5
5 4 3
```

### 样例 1 输出

```
3
2
4
15
-1
-1
```

样例 1 解释

对于每组数据，如下是满足条件的  $a_i$

- {3, 3, 3}
- {2, 2, 2, 2}
- {2, 3, 3, 3, 4, 4}
- {15}
- 不存在
- 不存在

样例 2 输入/输出

见下发文件中的 *bar2.in/ans*, 此样例满足测试点 20 的限制。

数据范围与约定

对于所有数据，满足  $T \in [1, 500]$ ;  $N \in [1, 2^{60} - 1]$ ;  $S, X \in [0, 2^{60} - 1]$ 。

测试点 1 – 10 按点得分，且每个测试点均独立于其他测试点。

测试点 11 – 20 捆绑测试，且依赖测试点 1 – 10。

测试点编号	约束
1, 2	$N, S, X, T \leq 5$
3, 4, 5	$N, S, X, T \leq 100$
6, 7	$T = 1$
8, 9, 10	$S = X$
11, 12, 13, 14, 15, 16, 17, 18, 19, 20	无特殊限制

# 三明治 (sandwich)

星垂平野阔，月涌大江流。——杜甫《旅夜书怀》

## 【题目描述】

请注意数据范围中的一些性质。

在二维平面上，有  $N$  块面包和  $N$  块肉饼，记第  $i$  块面包的位置为  $(a_i^x, a_i^y)$ ，记其美味度为  $a_i^w$ ；同理，记第  $i$  块肉饼的位置为  $(b_i^x, b_i^y)$ ，其美味度为  $b_i^w$ 。

假设你需要制作  $Q$  个三明治，第  $i$  个三明治所选材料的限制用一个整数对  $L_i, R_i$  表示，对于第  $i$  个三明治，需要选择一块面包  $j$  和一块肉饼  $k$ ，在同时满足如下条件时，最大化美味度之和，即求  $a_j^w + b_k^w$  的最大值。

- $a_j^y < b_k^y$
- $(a_j^x < L_i \wedge R_i < b_k^x) \vee (L_i < a_j^x \wedge b_k^x < R_i)$

其中  $\wedge$  表示逻辑与运算， $\vee$  表示逻辑或运算。

注意，在制作后所选择的面包和肉饼不会消失，也就是询问之间相互独立。

## 【输入格式】

第一行一个整数  $N$ 。

接下来  $N$  行，每行 3 个整数，表示  $a_i^x, a_i^y, a_i^w$ 。

接下来  $N$  行，每行 3 个整数，表示  $b_i^x, b_i^y, b_i^w$ 。

接下来一行一个整数  $Q$ 。

接下来  $Q$  行，每行两个整数  $L_i, R_i$ 。

## 【输出格式】

输出  $Q$  行，每行一个整数，表示所求的  $a_j^w + b_k^w$  的最大值，特别的，如果不存在合法的  $j, k$ ，输出  $-1$ 。

## 【样例】

### 样例 1 输入

```
2
-3 1 1
-6 3 10
3 4 100
5 2 1000
5
-5 4
-2 6
-4 1
-10 10
-1 2
```



样例 1 输出

```
101
-1
110
1001
1001
```

样例 1 解释

对于每个询问，如下是满足条件的  $(j, k)$

- $(1, 3)$
- 不存在
- $(2, 3)$
- $(1, 4)$
- $(1, 4)$

样例 2 输入/输出

见下发文件中的 *sandwich2.in/ans*，此样例满足子任务 1 的限制。

样例 3 输入/输出

见下发文件中的 *sandwich3.in/ans*，此样例满足子任务 6 的限制，且有  $L_1 = -10^9, R_1 = 10^9$ 。

数据范围与约定

对于所有数据，满足：

$N \in [1, 10^5]$   
 $Q \in [1, 5 \times 10^5]$   
 $a_i^x, L_i \in [-10^9, -1]; b_i^x, R_i \in [1, 10^9]$   
 $a_i^y, b_i^y \in [1, 10^9]$   
 $a_i^w, b_i^w \in [1, 10^9]$   
 $a_1^x, \dots, a_n^x, b_1^x, \dots, b_n^x, L_1, \dots, L_Q, R_1, \dots, R_Q$  两两不同  
 $a_1^y, \dots, a_n^y, b_1^y, \dots, b_n^y$  两两不同

子任务编号	分值	约束	依赖的子任务
1	10	$N, Q \leq 200$	无
2	15	$N, Q \leq 2000$	1
3	5	对于所有 $(i, j)$ , 有 $a_i^y < b_j^y$	无
4	10	$L_i = -10^9, R_i = 10^9$	无
5	30	$N, Q \leq 5 \times 10^4$	1, 2
6	30	无特殊限制	1, 2, 3, 4, 5