

# BCSP初中组编程能力题目样卷

题目名称	数列变换	歌词匹配	棋盘删数游戏	小 Z 的背包
题目类型	传统型	传统型	传统型	传统型
目录	array	match	game	backpack
可执行文件名	array	match	game	backpack
输入文件名	array.in	match.in	game.in	backpack.in
输出文件名	array.out	match.out	game.out	backpack.out
每个测试点时限	1.0 秒	1.0 秒	1.0 秒	2.0 秒
内存限制	256 MiB	512 MiB	512 MiB	512 MiB
测试点数目	20	10	10	20
测试点是否等分	是	是	是	否

## 提交源程序文件名

对于 C++ 语言	array.cpp	match.cpp	game.cpp	backpack.cpp
-----------	-----------	-----------	----------	--------------

## 编译选项

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

## T1 数列变换（array）

### 题目描述

小 Z 最初有一个长度为  $n$  的数列，第  $i$  个数为  $a_i$ 。现在小 Z 对数列进行了  $q$  次操作，对于每次操作有如下两种形式：

- 1 x：表示向数列中添加一个数  $x$ 。
- 2 x：表示询问数列重排序后第一个大于  $x$  的数是什么，如果没有，则输出  $-1$ 。

### 输入格式

从 `array.in` 文件读入数据。

第一行输入一个正整数  $n$ ，代表最初数列的长度。

第二行输入  $n$  个非负整数，第  $i$  个数为  $a_i$ ，两个数以空格隔开。

接下来输入一个正整数  $q$ ，表示小 Z 操作的次数。

接下来  $q$  行，每行两个以空格隔开的整数 `op x`：

- 1. 若 `op == 1`，表示向数列中添加一个非负整数  $x$ ；
- 2. 若 `op == 2`，表示询问数列中第一个大于  $x$  的数是什么。

## 输出格式

输出到 `array.out` 文件。

对于所有 `op == 2` 的询问，输出此时数列中第一个大于  $x$  的数是什么，如果没有，则输出 `-1`。

## 样例

### 输入数据1

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

### 输出数据1

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

## 说明/提示

- 对于 20% 的数据， $1 \leq n, q \leq 100$ ；
- 对于另外 30% 的数据， $1 \leq n \leq 2000, 1 \leq q \leq 10^5$ ，其中 `op == 1` 的次数  $\leq 2000$ 。
- 对于 100% 的数据， $1 \leq n, q \leq 10^5, 0 \leq a_i, x \leq 10^9, 1 \leq op \leq 2$ 。

## T2 歌词匹配 (match)

### 题目描述

小 Z 和小 Y 在唱歌，他们唱的歌曲非常奇怪，歌词都是用 `abcd...` 这共 26 个小写字母拼起来的。

具体地，小 Z 唱歌的歌词为  $S$ ，小 Y 唱歌的歌词为  $T$ 。小 Z 会重复唱他的歌词  $N$  遍，记这  $N$  遍相同的歌词拼成的一整首歌曲为  $SS$ ，小 Y 会重复唱他的歌词  $M$  遍，记这  $M$  遍相同的歌词拼成的一整首歌曲为  $TT$ 。

小 Z 和小 Y 都是一秒钟唱一个词，如果在某一秒，他俩唱的词一样，我们就认为在这一秒他们的歌曲发生了一次匹配。现在，保证他俩唱完整个歌曲的时间是相同的，即  $SS$  和  $TT$  的长度相同。

现在问，当他俩唱完整首歌的时候，发生歌词匹配的次数是几次。

例如，小 Z 的歌词 "ab"，他会唱 3 遍，小 Y 的歌词为 "baa"，他会唱 2 遍。那么小 Z 的整首歌就是 "ababab"，小 Y 的整首歌就是 "baabaa"，在第 3, 4, 5 秒的时候发生了歌词配对，所以配对次数为 3。

## 输入格式

从 `match.in` 文件读入数据。

第一行输入两个正整数  $N, M$  分别表示小 Z 和小 Y 重复歌词的遍数。

第二行输入一个字符串  $S$ ，表示小 Z 唱的歌词。

第三行输入一个字符串  $T$ ，表示小 Y 唱歌歌词。

输入的字符串只包含小写字母。

## 输出格式

输出到 `match.out` 文件。

输出为一个整数，表示唱完整首歌时发生歌词匹配的次数。

## 样例

### 输入数据1

1	3 2
2	ab
3	baa

### 输出数据1

1	3
---	---

### 输入数据2

1	30 20
2	abbb
3	bbaabb

### 输出数据2

1	70
---	----

## 说明/提示

对于 40% 的数据， $|SS| \leq 10^5$ 。

对于再 30% 的数据， $|S|, |T| \leq 10$ 。

对于 100% 的数据， $1 \leq N, M \leq 10^9, |S|, |T| \leq 10^6$ 。

其中， $|S|$  表示歌词  $S$  的长度，并且数据保证小 Z 和小 Y 的整首歌曲时间相同，即有  $|SS| = |TT|$ 。

## T3 棋盘删数游戏 (game)

### 题目描述

小 Z 和小 Y 又发现了一个棋盘游戏。这个棋盘初始时可以看成是  $n$  行  $n$  列的矩阵方格，其中第  $i$  行第  $j$  列的小方格内有一个正整数  $a_{i,j}$ 。

小 Z 和小 Y 轮流进行游戏，轮到自己的回合时，如果棋盘的最后一行或者最后一列的和是偶数，那么该玩家就可以拿走这个棋盘矩阵中的最后一行或者最后一列。

- 例如：此时轮到小 Z 的回合，棋盘还剩下  $x$  行  $y$  列，并且第  $x$  行所有元素之和为偶数，那么小 Z 就可以拿走  $x$  行的所有格子，剩下  $x - 1$  行  $y$  列。

现在，由小 Z 开始第一个回合，小 Y 进行第二个回合，接着又是小 Z，就这样交替执行，并且假设小 Z 和小 Y 都是绝顶聪明的人，他们都采用最优的策略（最有利于自己获胜的策略）去进行每一个回合的游戏。

如果轮到某一个人的回合，但是他却无法拿走棋盘上的最后一行或者最后一列，那么游戏结束，这个回合的人就输了。

问，在小 Z 先手的情况下，是否有一个种策略可以必胜？

### 输入格式

从 `game.in` 文件读入数据。

第一行，输入一个整数  $t$  表示数据组数，每组数据相互独立互不影响。

对于每一组数据，第一行输入一个整数  $n$  表示棋盘的大小。

接下来有  $n$  行，每行有  $n$  个整数，第  $i$  行第  $j$  列的整数  $a_{i,j}$  表示棋盘对应格子中的数。

### 输出格式

输出到 `game.out` 文件。

输出共  $t$  行，如果小 Z 可以找到一个必胜的策略，则输出 `W`，否则输出 `L`。

### 样例

#### 输入数据1

1	2
2	2
3	2 4
4	4 8
5	3
6	5 4 2
7	1 5 9
8	7 3 8

## 输出数据1

1	L
2	W

## 说明/提示

### 样例解释

对于第一组数据：小 Z 无论删除最后一行还是最后一列，都会留下一个 2 给小 Y，小 Y 可以删除这个 2，然后轮到小 Z，小 Z 发现棋盘空了，没有格子可以删除，就输了。

对于第二组数据：小 Z 只能删除最后一行（第 3 行），剩余棋盘第 2 行和为  $1 + 5 + 9 = 15$ ，第 3 列和为  $2 + 9 = 11$ ，发现无论是第 2 行还是第 3 列的和都变成了奇数，小 Y 无法拿走棋盘上的格子，小 Z 获胜。

### 数据范围

- 对于 30% 的数据，保证  $1 \leq n \leq 5$ 。
- 对于 50% 的数据，保证  $1 \leq n \leq 100$ 。
- 对于 70% 的数据，保证  $1 \leq n \leq 500$ 。
- 对于 100% 的数据，保证  $1 \leq n \leq 1000$ ， $1 \leq t \leq 5$ ， $1 \leq a_{i,j} \leq 2^{31} - 1$ 。

## T4 小 Z 的背包（backpack）

### 题目描述

小 Z 有一个体积为  $m$  的背包，现在有  $n$  件物品可以选择，第  $i$  件物品的体积为  $w_i$ ，价值为  $v_i$ ，每件物品只有一个。

小 Z 想要选择若干件物品放入背包，使得放入的物品体积之和在不超过背包体积  $m$  的情况下，物品的价值之和最大。

问，放入背包的物品价值和最大是多少？

### 输入格式

从 `backpack.in` 文件读入数据。

第一行输入两个整数  $n, m$  分别表示物品数量和背包容量。

接下来有  $n$  行，每行两个整数  $w_i, v_i$  分别表示第  $i$  件物品的体积和价值。

## 输出格式

输出到 `backpack.out` 文件。

一行一个整数表示答案。

## 样例

### 输入数据1

1	5	100
2	95	80
3	4	18
4	3	11
5	99	100
6	2	10

### 输出数据1

1	101
---	-----

## 说明/提示

### 样例解释

小 Z 可以选择第一个、第三个和第五个物品，体积和为 100，价值和为 101，是最大的。

### 数据范围

对于 20% 的数据，保证  $w_i \leq 1500$ 。

对于另外 30% 的数据，保证  $v_i \leq 1500$ 。

对于 100% 的数据，保证  $2 \leq n \leq 40, 0 \leq m \leq 10^{18}, 0 \leq w_i, v_i \leq 10^{15}$ 。