

安徽省大学生程序设计大赛

(本科组)

比

赛

试

题

2020 年 10 月

目 录

A. 数字排列	3
B. 嵌套线段.....	4
C. 前后相加	5
D. 字符串修改.....	7
E. 收集圣物.....	9
F. 跳蛙出行.....	12
G. 序列游戏	13
H. 字幕校对	15
I. 美丽几何	17
J. 飞奔的战士.....	18
K. 农夫打狼.....	20

A. 数字排列

给出两个整数 A 和 B ，可以重新排列 A 得到新的数字 C （不能有前导 0）。求在小于等于 B 的情况下， C 的最大值是多少。如果不存在输出 -1

输入说明

第一行包含两个整数 A 和 B 。 $1 \leq A, B < 10^9$

输出说明

输出符合条件情况下 C 的最大值

输入样例 1

1234 3456

输出样例 1

3421

输入样例 2

10000 5

输出样例 2

-1

B. 嵌套线段

给出 N 个线段, 对于线段 i , 给出线段在数轴上的左端点 l_i 和右端点 r_i , 对于线段 i 和 j , 假如 $l_j < l_i$ 并且 $r_i < r_j$ 就说明线段 i 嵌套在线段 j 中。找到所有嵌套在至少一个其他线段中的线段

输入说明

第一行是一个整数 N , $1 \leq N \leq 100$

接下来 N 行, 每行两个整数 l_i, r_i 表示第 i 个线段的端点。

输出说明

所有嵌套在其它线段中的线段个数。

输入样例

4

0 5

2 6

3 4

0 7

输出样例

2

说明: (3, 4) 和 (2, 6) 包含在 (0, 7) 中。

C. 前后相加

给定一个序列，长度为 N ，开始时，上面所有元素为 0。

你可以对序列作如下两种操作：

1. 指定一个整数 k ($1 \leq k \leq N$) 和一个**非降序列** $c_1, c_2, c_3, \dots, c_k$, (c_i 非负, $1 \leq i \leq k$)，对序列 x 的前 k 个数，令 $x_i = c_i + x_i$ 。

2. 指定一个整数 k ($1 \leq k \leq N$) 和一个**非升序列** $c_1, c_2, c_3, \dots, c_k$, (c_i 非负, $1 \leq i \leq k$)，对序列 x 的后 k 个数，令 $x_{[N-k+i]} = c_i + x_{[N-k+i]}$ 。

你的目标是将序列 x 构造为与序列 A 相等的序列，即 $x_i = A_i$ ($1 \leq i \leq n$)，输出最少需要多少此操作，以达成目标。

数据范围

$$1 \leq N \leq 2 * 10^5$$

$$1 \leq A_i \leq 10^9$$

输入说明

第一行为一个整数 N ，序列的长度。

第二行为 N 个整数，表示目标序列 A 。

输出说明

输出最少的操作次数。

样例一

输入

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

输出

```
3
```

样例二

输入

5

2 1 2 1 2

输出

2

D. 字符串修改

对于两个由 0 与 1 组成的长度均为 N 的序列 S 和 T ， $f(S, T)$ 定义如下：

重复如下操作，使得序列 S 与序列 T 相同。 $f(S, T)$ 是经过操作后，所需要的最少的花费。

改变 $S[i]$ 的值 (0 改为 1 或者 1 改为 0)，它的花费为 $D * C[i]$ ，其中 D 是在此操作之前，序列中满足 $S[j] \neq T[j]$ 的 j 的个数 ($1 \leq j \leq N$)。 $C[i]$ 表示对 i 位置进行变换的代价。

对于一个固定长度 N ，共有 $2^N * (2^N - 1)$ 对由 01 组成的序列 (S, T) ，计算所有 $f(S, T)$ 的总和，并输出它对 $10^9 + 7$ 的结果。

数据范围

$$1 \leq N \leq 2 * 10^5$$

$$1 \leq C[i] \leq 10^9$$

输入说明

输入第一行为一个整数 N ，表示 S 与 T 的长度，第二行共 N 个整数， $C[1], C[2], C[3], \dots, C[N]$ 。

输出说明

输出一个整数，所有 $f(S, T)$ 之和对 $10^9 + 7$ 取模的结果。

样例一

输入样例

```
1
1000000000
```

输出样例

```
999999993
```

样例二

输入样例

```
2
5 8
```

输出样例

```
124
```

提示

对样例 1 , S 可取 0、1 , T 可取 0、1 , 共有 $f(0,0)$, $f(0,1)$, $f(1,0)$, $f(1,1)$ 四种情况
 $f(0,0)=0, f(1,1)=0, f(1,0)=f(0,1)=10^9$, 故最终结果为 2×10^9 , 取模可得 999999993。

样例 2 中 , 针对 $S=(0,1), T=(1,0)$ 这个序列对 , 首先将 S_1 改为 1 , 造成的花费为 $5 \times 2=10$,
因为 S_1 与 T_1 , S_2 与 T_2 均不相同。再将 S_2 改为 0 , 造成的花费为 $8 \times 1=8$, 故 $f(S,T)=18$ 。

E. 收集圣物

在一个策略游戏中，僧侣单位可以收集圣物，放入修道院中，以生产黄金。因此，圣物对赢得游戏的胜利很重要。

某个僧侣正处于一个长为 n 宽为 m 的矩形迷宫中，迷宫的四周都是墙壁，不可移动。迷宫中圣物可能有若干个，为了降低游戏难度，僧侣只需要发现任意一个圣物，就能完成任务。

迷宫有若干扇门。同时，在迷宫的一些地方会有钥匙。每一把钥匙都只能使用一次，每一次可以开一扇门，即：一旦你通过一扇门后，门将自动关闭，如果想要再次打开这扇门，你需要再使用一把钥匙去打开它，通过之后门还将自动关闭。

输入这个迷宫，输出最快多少秒能够发现一个圣物。僧侣移速缓慢，每秒只能够向上下左右四个方向各移动一个单位。不过开门以及拾取钥匙不耗费时间。

如果僧侣无法收集任何圣物，输出“Bug Maze!!!”。

输入说明

第一行两个数 n, m ($n, m \leq 25$)。

接下来 n 行，每行 m 个字符，字符含义如下：

‘#’ -- 该位置为墙或者障碍。

‘.’ -- 该位置可以正常行走。

‘K’ -- 该位置有一把钥匙。

‘D’ -- 该位置有一扇门。

‘S’ -- 起始位置。‘S’ 仅有一个。

‘E’ -- 圣物位置。‘E’ 可能有多个。

数据保证钥匙的数量 ≤ 10 。

输出说明

仅一行，输出离开的最少时间。

如不能离开，输出“Bug Maze!!!”。

输入样例 1

5 6

##.S#E

##.###

##.K##

E##D##

##E.##

输出样例 1

7

输入样例 2

5 6

##.S#E

##.###

##.K##

E##D##

##ED##

输出样例 2

Bug Maze!!!

输入样例 3

5 7

###S###

##K...#

#KK##.#

#D##..#

#DE..##

输出样例 3

7

F. 跳蛙出行

池塘里有 n 片荷叶排成一行，有一只青蛙在上面跳跃。但是，这只青蛙是只不同寻常的青蛙，它每跳一次，只能从一片荷叶跳到相邻的荷叶上，并且，它在两片荷叶之间，只能跳跃有限次。青蛙可以从任意荷叶出发。问它最多能跳多少次。

输入说明

第一行为一个整数 n ，表示荷叶数。

第二行为 $n-1$ 个整数，第 i 个整数表示在第 i 片与第 $i+1$ 片荷叶之间，青蛙最多跳多少次（无论是从左到右，还是从右到左，均包括在内）。

输出说明

一个整数，表示跳蛙最多跳跃的次数。

输入样例

5

2 1 2 1

输出样例

5

提示：从第 3 片出发，按 3->4->3->2->1->2 的次序跳跃。

G. 序列游戏

有一个序列 w ，初始为空。再给出一个长度为 m 单调递增的序列 a 。你需要对序列 w 作如下 n 次操作：

(1) 操作 0，在序列尾部添加数字 0。

(2) 操作 1，在序列尾部添加数字 1。

(3) 操作 -1，删除序列 w 中，所有位于位置 a_i 的数 ($1 \leq i \leq m$)。比如 $a = \{1, 3, 5\}$ ，就将 w 中第 1, 3, 5 个数删除。若 $a_i > w$ 的当前长度，则该操作停止。

输出 n 次操作后的序列 w 。

输入说明

第一行为两个数， n 与 m ，表示操作数与 a 的长度， $1 \leq n, m \leq 10^6$ 。

第二行为 m 个数，表示序列 a ， $a_i \leq 10^6$ 。

后面共有 n 行，每行有一个整数 (0, 1, -1)，表示操作类型。

输出说明

最终的序列 w 。若序列为空，输出 Poor stack!

样例 1

输入样例

```
10 3
1 3 6
-1
1
1
0
0
-1
0
```

1

-1

1

输出样例

011

样例 2

输入样例

2 1

1

1

-1

输出样例

Poor stack!

H. 字幕校对

伴随着中国国产影视作品风靡国外,听不懂汉语的观众需要英文字幕的帮忙才能理解影片故事情节。现在小喇叭接到了这样一个任务,他需要按照标准.STA字幕格式的需要,对字幕文件进行编辑。标准英文字母格式如下:

[Ordinal number of the block]

[Beginning time of the subtitles --> Ending time of the subtitles]

[Subtitle text in one or more lines]

每两个字幕信息之间用一行空行隔开。时间格式为HH:MM:SS, TTT(注意是英文的冒号和逗号),表示 小时:分钟:秒,毫秒。

这是一个.STA格式文件的样例:

1

00:00:01,600 --> 00:00:04,200

Good day!

2

00:00:05,900 --> 00:00:07,999

Good day to you too!

Here you go!

3

00:00:10,000 --> 00:00:14,000

May I please have ten garlic sausages?

由于影片播放时需要对原有视频进行编辑以适应播出时长,所以字母文件也

要做适当的调整，因此小喇叭的任务就是编写程序完成对字幕文件的编辑。

输入说明

每个测试样例最多不超过30行。第一行是字幕序号，第2行是按照. STA字幕格式的时间信息，第3行开始时字母信息，可能有多行字幕。为了便于处理，字幕中仅包含大小写字母和如下符号 ' , . ? ! 。 (' 。 ' 不含在内)

最后一个字幕信息块后是一个'#' 号，表明本测试样例末尾。

'#' 号之后是一个整数T ($-10000 \leq T \leq 10000$) , 表示需要对字幕文件中时间信息进行调整的偏移量。

输出说明

按照给出的T（毫秒）编辑需求，调整字母文件中的时间信息（每个字幕块中的开始和结束时间）。并将修改好时间的字幕文件打印出来。测试数据保证编辑后的时间不会出现负值。

输入样例

```
8
00:00:01,600 --> 00:00:04,200
We thought you was...
9
00:00:05,900 --> 00:00:07,999
a toad.
#
300
```

输出样例

```
8
00:00:01,900 --> 00:00:04,500
We thought you was...
9
00:00:06,200 --> 00:00:08,299
a toad.
#
```


I. 美丽几何

在平面上有 n 个点，初始每个点的美丽值都为 0，任意选择两个点组成一条直线，对于每一条直线，如果存在一个点，这个点到这条直线的距离小于其他 $n-3$ 个点到这条直线的距离，那么我们把这个点的美丽值加 1。为了简化输出，我们只需要输出所有点的美丽值的异或值，保证三点不共线。

输入说明

第一行一个正整数 n ($4 \leq n \leq 2000$)

接下来 n 行，每一行有 2 个正整数 x, y 。代表一个点的坐标 ($0 \leq x, y \leq 100\,000\,000$)

输出说明

输出所有点的美丽值的异或值。

输入样例

```
4
0 0
0 1
1 0
1 1
```

```
4
0 0
1 0
1 2
2 1
```

输出样例

```
0

3
```

J. 飞奔的战士

众所周知，Teutonic Knight 是某 RTS 游戏中，移动速度最快的单位。而且他热衷于参加赛跑比赛。

现在他想知道他最快需要用多长时间才可以到达终点，假设他没有开加速挂。

起初，他站在第 1 张图的某个位置（用‘S’标识），终点在第 k 张图的某个位置（用‘E’标识），在除最后一张图外，每一张图上有且仅有一个传送点（用‘T’标识），可以将他传送到下一张图的某个位置，用‘t’标识。传送不需要任何时间，可以理解为瞬间完成。每次移动一个单位距离耗时 1 秒。

现在 Teutonic Knight 想知道他在全程狂奔的情况下，最快需要多长时间才能到达终点。

输入说明

第一行三个数 k, n, m ($k, n, m \leq 100$)。

接下来 k 个图，两个图之间用一个空行隔开，每个图有 n 行，每行 m 个字符，字符含义如下：

‘#’ -- 该位置为墙或者障碍。

‘.’ -- 该位置可以正常行走。

‘T’ -- 该位置有一传送点。

‘t’ -- 从上图传送来的位置。

‘S’ -- Teutonic Knight 的起始位置。‘S’ 仅有一个。

‘E’ -- 出口。‘E’ 仅有一个。

输出说明

仅一行，输出到达终点的最少时间。

如不能离开，输出 “Trapped Maze!!!”。

输入样例

```
3 5 6
##.S#.
##.###
##....
.##.#.
##..#T

##.t..
##.##.
##.#..
T.#..#
#...##

###t##
##..#E
#..##.
#.#.#.
#....#
```

输出样例

```
33
```

K. 农夫打狼

小明是某 RTS 游戏的狂热爱好者，但是他从来不好好打游戏，经常搞出一些令人窒息的操作。一次，他竟然将自己的一个农夫派出去打野狼。

农夫初始时站在地图坐标 $(0, 0)$ 的位置，地图的出口的坐标为 (n, n) ，地图上共有 m 只野狼，每只野狼有一个坐标 (x_i, y_i) ，数据保证不会有两个野狼在相同的位置。农夫只有消灭地图上所有的野狼才能完成任务。

整个地图按 y 轴一共分为 n 层，当且仅当农夫清掉了本层的所有怪物之后才能到下一层，比如说，如果农夫当前的位置是 $(x, 3)$ ，那么他必须将 y 坐标等于 3 的所有野狼消灭，才允许走到 $(x, 4)$ 。如果本层没有怪物，农夫能够直接到下一层。

由于小明使用的种族是 Magyar，这个种族骁勇善战，农夫只需一击就能将野狼打死。因而，每到一个位置，如果这个位置有野狼，他可以直接将怪物秒掉（不需要任何操作）。

现在给出 m 个野狼的坐标，问农夫最少需要走多少步才能完成任务。农夫向上下左右移动一格算一步。

输入说明

第一行两个数 n, m ($n, m \leq 100,000$)。

接下来 m 行，每行两个数，代表第 i 只野狼的坐标。

输出说明

仅一行，输出农夫最少需要走多少步。

输入样例

4 5

1 1

2 1

1 2

3 2

3 3

输出样例

10