

FCS2018 模拟试题

ExfJoe

February 9, 2018

竞赛时长: 240min

试题名称	红绿灯	全排列	点点井井
可执行文件名	light	perm	square
输入文件名	light.in	perm.in	square.in
输出文件名	light.out	perm.out	square.out
时间限制	1s	1s	1s
空间限制	256M	512M	256M
测试点数目	10	10	20
测试点分数	10	10	5
是否有 SPJ	否	否	否
题目类型	传统	传统	传统

- 认真独立完成试题，不与他人交流讨论
- 最终评测使用 Lemon，默认栈空间限制为 8M，开启 O2 优化
- 试题按英文名称字典序排序

红绿灯

题目描述

泰迪每天都要通过一条路从家到学校，这条路的起点是泰迪家，终点则是学校。

这条路中间还有 n 个路口，从第 $i-1$ 个路口走到第 i 个路口需要 d_i 秒，每个路口都有一个红绿灯。更具体地，绿灯持续时间是 g 秒，红灯持续时间是 r 秒。每天从第 0 秒开始，所有灯都是绿灯，持续 g 秒之后变为红灯，再过 r 秒变成绿灯，以此类推，并且同一时刻所有灯都是相同状态。当泰迪到达一个路口，若是绿灯则可直接通过，若是红灯则需原地等待至绿灯。若到达某一路口时灯的状态正好发生改变，则视达到路口时灯的颜色为其改变后的颜色，例如第 g 秒到达一个路口则视为遇到红灯。

现在泰迪预计了接下来 q 天从家出发的时间，第 j 天将会在第 t_j 秒从家出发，他希望你告诉他每天到达学校的最早时间。你可以假定一天内泰迪一定可以到达学校。

输入格式

第一行三个正整数 n, g, r 表示路口数以及绿灯、红灯持续的时间。

第二行 $n+1$ 个正整数 d_i 表示相邻路口间的通行时间， d_1 表示从起点到第一个路口的所需时间， d_{n+1} 表示第 n 个路口到终点的所需时间。

第三行一个正整数 q 表示询问天数。

第四行 q 个非负整数 t_j 表示每天的出发时间。

输出格式

共 q 行，对于每次询问输出一行一个整数表示答案。

样例

Input	
1 3 2	
5 2	
5	
1 2 3 4 5	
Output	
8	
9	
12	
12	
12	

数据范围

30% 的数据： $n, q \leq 1000$

另有 20% 的数据： $g, r \leq 100$

另有 20% 的数据： d_i 均相同

100% 的数据： $n, q \leq 5 \times 10^4$, $2 \leq g, r \leq 10^9$, $d_i, t_j \leq 10^9$

全排列

题目描述

定义两个长为 n 的排列 A 与 B 相似：若 $\forall i$ ，满足 $C(A, A_i) = C(B, B_i)$ 。其中 $C(P, x)$ 为满足 $P_j < x (1 \leq j \leq n)$ 的 j 的数目。

对于两个长为 n 的排列 P_1, P_2 ，定义函数 $F(P_1, P_2)$ 等于满足 $P_1[l \dots r]$ 相似于 $P_2[l \dots r] (1 \leq l \leq r \leq n)$ 并且 $P_1[l \dots r]$ 包含不超过 E 个逆序对的数对 (l, r) 的数目。

现在请你求出：对 P_1, P_2 分别取遍所有 $1 \sim n$ 的排列后所有 $F(P_1, P_2)$ 的和。

输入格式

第一行一个整数 T 表示数据组数。

接下来 T 行每行两个非负整数 n, E 。

输出格式

对于每组数据输出一行一个整数表示答案，答案对 $10^9 + 7$ 取模。

样例

Input
4
2 2
2 1
2 0
1 1

Output
10
10
9
1

数据范围

30% 的数据： $T \leq 10, n \leq 5, E \leq 10$

50% 的数据： $T \leq 10^4, n \leq 10, E \leq 50$

80% 的数据： $T \leq 10^4, n \leq 50, E \leq 10^6$

100% 的数据： $T \leq 10^4, n \leq 500, E \leq 10^6$

点点井井

题目描述

在一个二维平面上有一个图形，如下图所示：

```
#####  
#.....#  
#.#####.#  
#.#.....#.#  
#.#.#####.#.#  
#.#.#.....#.#.#  
#.#.#.###.#.#.#  
#.#.#.#.#.#.#.#  
#.#.#.###.#.#.#  
#.#.#.....#.#.#  
#.#.#####.#.#  
#.#.....#.#  
#.#####.#  
#.....#  
#####
```

它是一个中心为一个字符'.'，之后按照一层'#' 一层'.' 的顺序围起来的无限大的图形。

现在给定若干个位置上的字符，请你确定出这个图形中心位置的坐标，若有多个合法中心位置，则输出离原点曼哈顿距离最近的一个，若仍有多个，则输出横坐标最大的一个，若还有多个，输出纵坐标最大的一个；若没有合法中心点，输出"Too damaged"。

输入格式

第一行一个正整数 T 表示数据组数。

对于每组数据，第一行一个正整数 n 代表已知位置数。

接下来 n 行，每行两个整数 x, y 和一个字符 c ，表示该位置的坐标以及该位置上的字符。

输出格式

对于每组数据输出一行表示答案。格式为："Case #x: ans"，其中 x 为当前数据的组编号，ans 表示答案，# 之前有一个空格，: 之后有一个空格，除此之外不允许有任何多余字符。

样例

Input	
6	
1	
0 0 .	
1	
0 0 #	
3	
0 0 #	
0 1 #	
1 0 #	
5	
50 30 #	
49 30 #	

```
49 31 #
49 32 #
50 32 #
2
-98 0 #
99 50 .
4
88 88 .
88 89 .
89 88 .
89 89 .
```

Output

```
Case #1: 0 0
Case #2: 1 0
Case #3: 1 1
Case #4: 50 31
Case #5: 1 0
Case #6: Too damaged
```

数据范围

30% 的数据: $n \leq 100$, $|x|, |y| \leq 100$

100% 的数据: $n \leq 1000$, $|x|, |y| \leq 10^{15}$, $T \leq 50$, $c \in \{., \#\}$