苏州市人工智能学会

青少年人工智能核心算法素养考核(SACCC)

6级

时间：202X年X月X日 X:00~X:00

题目名称	积木大赛	报数	时间复杂度	策略游戏
题目类型	传统型	传统型	传统型	传统型
目录	block	number	complexity	game
可执行文件名	block	number	complexity	game
输入文件名	block.in	number.in	complexity.in	game.in
输出文件名	block.out	number.out	complexity.out	game.out
时间限制	1.0 s	1.0 s	1.0 s	1.0 s
内存限制	128 MB	256 MB	256 MB	256 MB
测试点数目	10	10	10	20

1　积木大赛

题目描述

题目描述

苏州的拙政园举办了一年一度的“园林积木大赛”。今年比赛的内容是搭建一座宽度为 n 的园林景观，景观可以看成由 n 块宽度为 1 的积木组成，第 i 块积木的最终高度需要是 h_i。

在搭建开始之前，没有任何积木（可以看成 n 块高度为 0 的积木）。接下来每次操作，小朋友们可以选择一段连续区间 $[l,r]$，然后将第 L 块到第 R 块之间（含第 L 块和第 R 块）所有积木的高度分别增加 1。

小 M 是个聪明的小朋友，她很快想出了建造园林景观的最佳策略，使得建造所需的操作次数最少。但她不是一个勤于动手的孩子，所以想请你帮忙实现这个策略，并求出最少的操作次数。

输入格式

包含两行，第一行包含一个整数 n，表示大厦的宽度。

第二行包含 n 个整数，第 i 个整数为 h_i。

输出格式

建造所需的最少操作数。

输入输出样例 #1

输入 #1

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

输出 #1

```
5
```

说明/提示

样例解释

其中一种可行的最佳方案，依次选择：$[1,5]$，$[1,3]$，$[2,3]$，$[3,3]$，$[5,5]$。

数据范围

- 对于 30% 的数据，有 $1 \le n \le 10$；
- 对于 70% 的数据，有 $1 \le n \le 1000$；
- 对于 100% 的数据，有 $1 \le n \le 100000$，$0 \le h_i \le 10000$。

2　报数

题目描述

报数游戏是一个广为流传的休闲小游戏。参加游戏的每个人要按一定顺序轮流报数，但如果下一个报的数是 7 的倍数，或十进制表示中含有数字 7，就必须跳过这个数，否则就输掉了游戏。

在一个风和日丽的下午，小 r 和小 z 闲得无聊玩起了这个报数游戏。但在只有两个人玩的情况下计算起来还是比较容易的，因此他们玩了很久也没分出胜负。此时小 z 灵光一闪，决定把这个游戏加强：任何一个十进制中含有数字 7 的数，它的所有倍数都不能报出来！

形式化地，设 $p(x)$ 表示 x 的十进制表示中是否含有数字 7，若含有则 $p(x)=1$，否则 $p(x)=0$。则一个正整数 x 不能被报出，当且仅当存在正整数 y 和 z，使得 $x=yz$ 且 $p(y)=1$。

例如，如果小 r 报出了 6，由于 7 不能报，所以小 z 下一个需要报 8；如果小 r 报出了 33，则由于 $34=17\times 2$，$35=7\times 5$ 都不能报，小 z 下一个需要报出 36；如果小 r 报出了 69，由于 $70\sim 79$ 的数都含有 7，小 z 下一个需要报出 80 才行。

现在小 r 的上一个数报出了 x，小 z 想快速算出他下一个数要报多少，不过他很快就发现这个游戏可比原版的游戏难算多了，于是他需要你的帮助。当然，如果小 r 报出的 x 本身是不能报出的，你也要快速反应过来小 r 输了才行。

由于小 r 和小 z 玩了很长时间游戏，你也需要回答小 z 的很多个问题。

输入格式

第一行，一个正整数 T 表示小 z 询问的数量。

接下来 T 行，每行一个正整数 x，表示这一次小 r 报出的数。

输出格式

输出共 T 行，每行一个整数，如果小 r 这一次报出的数是不能报出的，输出 -1，否则输出小 z 下一次报出的数是多少。

输入输出样例 #1

输入 #1

```
4
6
33
69
300
```

输出 #1

```
8
36
80
-1
```

输入输出样例 #2

输入 #2

```
5
90
99
106
114
169
```

输出 #2

```
92
100
109
-1
180
```

输入输出样例 #3

输入 #3

```
见附件中的 number/number3.in
```

输出 #3

```
见附件中的 number/number3.ans
```

输入输出样例 #4

输入 #4

```
见附件中的 number/number4.in
```

输出 #4

```
见附件中的 number/number4.ans
```

说明/提示

【样例解释 #1】

这一组样例的前 3 次询问在题目描述中已有解释。

对于第 4 次询问，由于 $300 = 75 \times 4$，而 75 中含有 7，所以小 r 直接输掉了游戏。

【数据范围】

对于 10% 的数据，$T \leq 10$，$x \leq 100$。
对于 30% 的数据，$T \leq 100$，$x \leq 1000$。
对于 50% 的数据，$T \leq 1000$，$x \leq 10000$。
对于 70% 的数据，$T \leq 10000$，$x \leq 2 \times 10^5$。
对于 100% 的数据，$1 \leq T \leq 2 \times 10^5$，$1 \leq x \leq 10^7$。

3 时间复杂度

题目描述

小明正在学习一种新的编程语言 A++，刚学会循环语句的他激动地写了好多程序并给出了他自己算出的时间复杂度，可他的编程老师实在不想一个一个检查小明的程序，于是你的机会来啦！下面请你编写程序来判断小明对他的每个程序给出的时间复杂度是否正确。

A++语言的循环结构如下：

```
F i x y
    循环体
E
```

其中`F i x y`表示新建变量 i（变量 i 不可与未被销毁的变量重名）并初始化为 x， 然后判断 i 和 y 的大小关系，若 i 小于等于 y 则进入循环，否则不进入。每次循环结束后 i 都会被修改成 $i+1$，一旦 i 大于 y 终止循环。

x 和 y 可以是正整数（x 和 y 的大小关系不定）或变量 n。n 是一个表示数据规模的变量，在时间复杂度计算中需保留该变量而不能将其视为常数，该数远大于 100。

`E` 表示循环体结束。循环体结束时，这个循环体新建的变量也被销毁。

注：本题中为了书写方便，在描述复杂度时，使用大写英文字母 O 表示通常意义下 Θ 的概念。

输入格式

输入文件第一行一个正整数 t，表示有 t（$t \leq 10$）个程序需要计算时间复杂度。 每个程序我们只需抽取其中`F i x y`和`E`即可计算时间复杂度。注意：循环结构允许嵌套。

接下来每个程序的第一行包含一个正整数 L 和一个字符串，L 代表程序行数，字符串表示这个程序的复杂度，`O(1)`表示常数复杂度，`O(n^w)`表示复杂度为 n^w，其中 w 是一个小于 100 的正整数，输入保证复杂度只有`O(1)`和`O(n^w)`两种类型。

接下来 L 行代表程序中循环结构中的`F i x y`或者`E`。 程序行若以`F`开头，表示进入一个循环，之后有空格分离的三个字符（串）`i x y`， 其中 i 是一个小写字母（保证不为n），表示新建的变量名，x 和 y 可能是正整数或 n， 已知若为正整数则一定小于 100。

程序行若以`E`开头，则表示循环体结束。

输出格式

输出文件共 t 行，对应输入的 t 个程序，每行输出`Yes`或`No`或者`ERR`，若程序实际复杂度与输入给出的复杂度一致则输出`Yes`，不一致则输出`No`，若程序有语法错误（其中语法错误只有:① `F`和`E`不匹配 ②新建的变量与已经存在但未被销毁的变量重复两种情况），则输出`ERR`。

注意：即使在程序不会执行的循环体中出现了语法错误也会编译错误，要输出`ERR`。

输入输出样例 #1

输入 #1

```
8
2 O(1)
F i 1 1
E
2 O(n^1)
F x 1 n
E
1 O(1)
F x 1 n
4 O(n^2)
F x 5 n
F y 10 n
E
E
4 O(n^2)
F x 9 n
E
F y 2 n
E
4 O(n^1)
F x 9 n
F y n 4
E
E
4 O(1)
F y n 4
F x 9 n
E
E
4 O(n^2)
F x 1 n
F x 1 10
E
E
```

输出 #1

```
Yes
Yes
ERR
Yes
No
Yes
Yes
ERR
```

说明/提示

【输入输出样例解释 1】

第一个程序 i 从 1 到 1 是常数复杂度。

第二个程序 x 从 1 到 n 是 n 的一次方的复杂度。

第三个程序有一个 `F` 开启循环却没有 `E` 结束，语法错误。

第四个程序二重循环，n 的平方的复杂度。

第五个程序两个一重循环，n 的一次方的复杂度。

第六个程序第一重循环正常，但第二重循环开始即终止（因为 n 远大于 100，100 大于 4）。

第七个程序第一重循环无法进入，故为常数复杂度。

第八个程序第二重循环中的变量 x 与第一重循环中的变量重复，出现语法错误②，输出 `ERR`。

【数据规模与约定】

对于 30% 的数据：不存在语法错误，数据保证小明给出的每个程序的前 $L/2$ 行一定为以 `F` 开头的语句，第 $L/2+1$ 行至第 L 行一定为以 `E` 开头的语句，$L \leq 10$，若 x、y 均为整数，x 一定小于 y，且只有 y 有可能为 n。

对于 50% 的数据：不存在语法错误，$L \leq 100$，且若 x、y 均为整数，x 一定小于 y， 且只有 y 有可能为 n。

对于 70% 的数据：不存在语法错误，$L \leq 100$。

对于 100% 的数据：$L \leq 100$。

4 策略游戏

题目描述

小 L 和小 Q 在玩一个策略游戏。

有一个长度为 n 的数组 A 和一个长度为 m 的数组 B，在此基础上定义一个大小为 $n \times m$ 的矩阵 C，满足 $C_{ij} = A_i \times B_j$。所有下标均从 1 开始。

游戏一共会进行 q 轮，在每一轮游戏中，会事先给出 4 个参数 l_1, r_1, l_2, r_2，满足 $1 \leq l_1 \leq r_1 \leq n$、$1 \leq l_2 \leq r_2 \leq m$。

游戏中，小 L 先选择一个 $l_1 \sim r_1$ 之间的下标 x，然后小 Q 选择一个 $l_2 \sim r_2$ 之间的下标 y。定义这一轮游戏中二人的得分是 C_{xy}。

小 L 的目标是使得这个得分尽可能大，小 Q 的目标是使得这个得分尽可能小。同时两人都是足够聪明的玩家，每次都会采用最优的策略。

请问：按照二人的最优策略，每轮游戏的得分分别是多少？

输入格式

第一行输入三个正整数 n, m, q，分别表示数组 A，数组 B 的长度和游戏轮数。

第二行：n 个整数，表示 A_i，分别表示数组 A 的元素。

第三行：m 个整数，表示 B_i，分别表示数组 B 的元素。

接下来 q 行，每行四个正整数，表示这一次游戏的 l_1, r_1, l_2, r_2。

输出格式

输出共 q 行，每行一个整数，分别表示每一轮游戏中，小 L 和小 Q 在最优策略下的得分。

输入输出样例 #1

输入 #1

```
3 2 2
0 1 -2
-3 4
1 3 1 2
2 3 2 2
```

输出 #1

```
0
4
```

输入输出样例 #2

输入 #2

```
6 4 5
3 -1 -2 1 2 0
1 2 -1 -3
1 6 1 4
1 5 1 4
1 4 1 2
2 6 3 4
2 5 2 3
```

输出 #2

```
0
-2
3
2
-1
```

说明/提示

【样例解释 #1】

这组数据中，矩阵 C 如下：

$$\begin{bmatrix} 0 & 0 \\ -3 & 4 \\ 6 & -8 \end{bmatrix}$$

在第一轮游戏中，无论小 L 选取的是 $x=2$ 还是 $x=3$，小 Q 都有办法选择某个 y 使得最终的得分为负数。因此小 L 选择 $x=1$ 是最优的，因为这样得分一定为 0。

而在第二轮游戏中，由于小 L 可以选 $x=2$，小 Q 只能选 $y=2$，如此得分为 4。

【样例 #3】

见附件中的 `game/game3.in` 与 `game/game3.ans`。

【样例 #4】

见附件中的 `game/game4.in` 与 `game/game4.ans`。

【数据范围】

对于所有数据，$1 \le n, m, q \le 10^5$，$-10^9 \le A_i, B_i \le 10^9$。对于每轮游戏而言，$1 \le l_1 \le r_1 \le n$，$1 \le l_2 \le r_2 \le m$。

测试点编号	$n, m, q \le$	特殊条件
1	200	$1, 2$
2	200	1
3	200	2
$4 \sim 5$	200	无
6	1000	$1, 2$
$7 \sim 8$	1000	1
$9 \sim 10$	1000	2
$11 \sim 12$	1000	无
13	10^5	$1, 2$
$14 \sim 15$	10^5	1
$16 \sim 17$	10^5	2
$18 \sim 20$	10^5	无

其中，特殊性质 1 为：保证 $A_i, B_i > 0$。
特殊性质 2 为：保证对于每轮游戏而言，要么 $l_1 = r_1$，要么 $l_2 = r_2$。