

NOIp信心膜你赛

出题人：zjp_shadow

题目名称	空间复杂度	统计正方形	最长不下降子序列
可执行文件名	complexity	squares	sequence
输入文件名	complexity.in	squares.in	sequence.in
输出文件名	complexity.out	squares.out	sequence.out
每个测试点时限	1s	1s	1s
内存限制	128MB	128MB	512MB
每个测试点分值	10	10	10
题目类型	传统	传统	传统

编译命令为 `!g++ % -o %< -lm` ，但似乎 LKOJ 自带 `-O2 -std=c++11` ，所以不作限制。

空间复杂度 (complexity)

题意描述

初语 正在学习一种新的编程语言 `B++`，刚学会结构体的他激动地写了好多程序并给出了他自己算出的空间复杂度，可他的编程老师实在不想一个一个检查 初语 的程序，于是你的机会来啦！下面请你编写程序来判断 初语 对他的每个程序给出的空间复杂度是否正确。

`B++` 语言的结构体如下：

```
S a
    b c
E
```

其中 `S` 表示声明一个结构体，`a` 表示结构体的名字，`b` 表示另外一个结构体的名字，`c` 表示 `b` 的维度（表示）。

然后会有一行单行的 `E` 表示结构体声明结束。

一开始会有一个标准类型 `i` 占用 $O(1)$ 的空间，我们自动认为，声明一个结构体就会占用它所需的空間。

比如：

```
S node
    i [n][200]
    i [n][n]
E
S edge
    node [n][2][n]
E
```

比如 `node` 会占用 $O(n^2)$ 的空间，那么 `edge` 会占用 $O(n^4)$ 的空间，所以这段代码最后占用 $O(n^4)$ 的空间，此处假设 n 远远大于常数。

现在给你一段共有 L 行的代码，需要你输出它的空间复杂度。

输入格式

单组数据，总共 L 行代码，每行代码都如同上面的形式，保证没有 `B++` 语法错误。

输出格式

输出一行 $O(n^w)$ 注意 $w = 0$ 时输出 $O(1)$ ， $w = 1$ 时输出 $O(n)$ 。

样例

样例一

样例输入：

```
S xsc
    i [n][666][n]
E
S orz
    xsc [2333][n][n]
    i [n][n][n][23333]
E
S chuyu
    i [888]
E
```

样例输出：

```
0(n^4)
```

样例二

样例输入：

```
S a
    i[123][456][789]
E
S b
    a[123][456][789]
E
```

样例输出：

```
0(1)
```

数据范围

对于所有数据 $L \leq 1000$ 。

保证每个结构体名字长度不超过 20 且全为小写字母，并且声明的 `[]` 个数不超过 20 （且里面的常量大小不超过 10^9 且不为 0 ），保证声明结构体中使用的类型在之前一定存在，并不会重复声明。

对于前 30% 的数据，不存在结构体嵌套的情况且保证结构体名字长度为 1 。

对于前 60% 的数据，每个结构体里仅存在一个类型。

对于 100% 的数据，不存在特殊限制。

统计正方形(squares)

题意描述

给你一个网格图，其中 x 坐标从 0 到 n ， y 坐标从 0 到 m 。（所以一共有 $(n+1)(m+1)$ 个点）

现在想让你告诉 茶叶盒 这个网格图上总共有多少个格点正方形。（也就是四个端点都在个网格图的交点上，且四边的长度都相等）

输入格式

总共一行，有两个正整数 n, m ，表示网格图的长和宽。

输出格式

输出一行答案，表示正方形个数。（注意答案对于 $10^9 + 7$ 取模）

样例

样例一

样例输入：

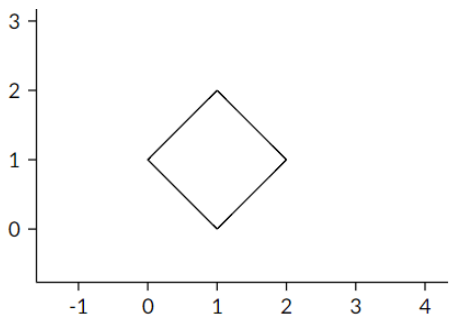
2 2

样例输出：

6

除了有 4 个 1×1 的正方形。

还有两个这样的下面这种形状正方形：



样例二

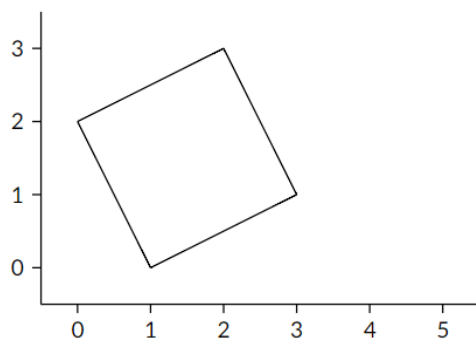
样例输入：

3 3

样例输出：

20

其中有一个这样的正方形：



样例三

样例输入：

233 66

样例输出：

10048258

数据范围

对于前 30% 的数据，满足 $n, m \leq 50$ ；

对于前 60% 的数据，满足 $n, m \leq 300$ ；

对于另外 10% 的数据，满足 $n = 1$ ；

对于另外 10% 的数据，满足 $n = 2$ ；

对于 100% 的数据，满足 $n, m \leq 10^6$ 。

最长不下降子序列(sequence)

题意描述

~~大动物联邦之主~~ 为了扫四合，决定给你一个长度为 n 的序列 $\{a_i\}$ 。

由于 ~~九彩幻神母鹿~~ 曾经是个卧底，常常反向打自己人，所以现在你可以选择任意一个区间 $[l, r]$ 并将区间内的数逆序。

~~万兽之王~~ 已经扫了三合，现在要爬山去攻打勇国，于是她问操作之后序列 $\{a_i\}$ 的最长不下降子序列长度。

由于他比较苟，区间只能被翻转一次。

输入格式

第一行一个整数 n 表示序列的长度。

然后一行共 n 个整数 a_i 。

输出格式

一行一个整数 ans 表示最长不下降子序列的长度。

样例

样例一

样例输入：

```
4
1 2 1 2
```

样例输出：

```
4
```

样例解释：

考虑将 $2 \rightarrow 3$ 翻转，那么整个序列就变成 $1, 1, 2, 2$ ，显然整个序列都是不下降的。

样例二

样例输入：

```
10
1 1 2 2 2 1 1 2 2 1
```

样例输出：

```
9
```

样例三

见下发样例 `sequence3.in/out` 。

数据范围

对于所有数据 $a_i \in \{1, 2\}$ 。

对于前 10% 的数据，满足 $a_i = 1$ 。

对于前 30% 的数据，满足 $n \leq 50$ 。

对于前 70% 的数据，满足 $n \leq 2000$ 。

对于 100% 的数据，满足 $n \leq 10^6$ 。