

7月10日模拟赛

代码命名：[题目英文名].cpp

输入/输出文件名：[题目英文名].in/.out

Problem A. 后缀表达式求值

题目英文名: calc

时空限制: 1s/256MB

题目描述:

每个四则运算算式都可以被写成一棵**算式二叉树**，算式二叉树的**叶子**节点放的是表达式中的数值，其他节点（保证它们**一定有两个子节点**）则放着算式中的运算符。

算式二叉树上，每个**节点的值**是：

- 如果这个节点上写的是数字，那么这个数字就是它的值。
- 否则（如果这个节点上写的是运算符 \oplus ），那么它一定有两个子节点；如果左子节点的值是 a ，右子节点的值是 b ，那么这个节点的值是 $a \oplus b$ 。

算式二叉树的值（也是它所代表的表达式的值）就是它的**根节点的值**。

为了避免除零等情况出现，本题中的运算符 \oplus 只会是 $+$ ， $-$ ， $*$ 的一种，其中 $*$ 表示乘法。

为了方便输入，所有存放数值的节点存放的都是 $0 \sim 9$ 之间的整数，因此输入中每一位数字都代表一个节点。

算式二叉树的后序遍历是一个字符串：

- 如果这个算式二叉树是空的，那么它的后序遍历也是空串。
- 否则，它的后续遍历是：根节点左子树的后序遍历，加上根节点右子树的后序遍历，加上代表根节点上的数字或运算符的字符。

现在给出一个算式二叉树的后序遍历，请你求出：

1. 有多少种可能的算式二叉树。
2. 所有可能的算式二叉树的值的和。

由于出题人不想写高精度，两问输出的值都对998244353取模。

输入格式:

输入一个字符串，表示算式二叉树的后序遍历。

输出格式:

第一行输出：有多少种可能的算式二叉树。

第二行输出：所有可能的算式二叉树的值的和。

两行输出的数都对998244353取模。

注意，取模后的答案范围为 $[0, 998244353)$ ，例如第二问的答案如果为 -1 ，你应该输出998244352。

样例输入：

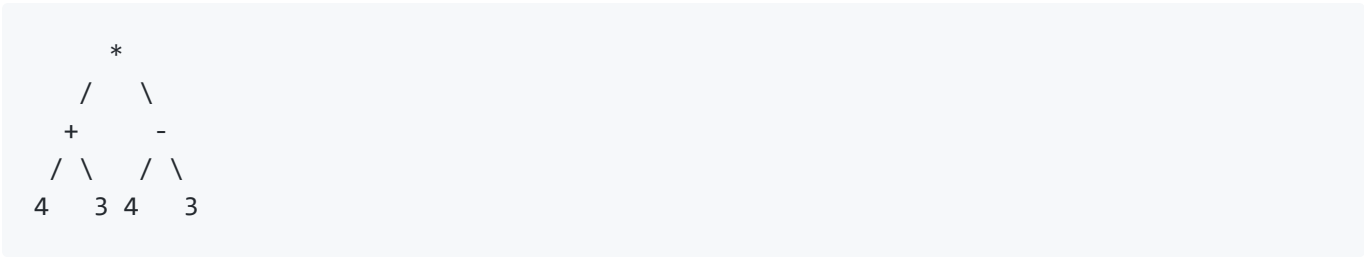
```
43+43-*
```

样例输出：

```
1
7
```

样例解释：

唯一的一种可能的二叉树如下：



它的值是7。

数据范围：

对于40%的数据：字符串长度不超过8

对于100%的数据：字符串长度不超过100，字符只有可能是 $+, -, *, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9$ 之一

Problem B. 矩阵字典序

题目英文名: matrix

时空限制: 1s/256MB

题目描述:

有一个 $n \times n$ 的二维矩阵 a , 你可以进行任意次 (包括0次) 操作, 每次操作选择一个整数 k 满足 $1 \leq k \leq n$, 交换第 k 行和第 k 列。具体来说, 对于 $1 \leq i \leq n$ 和你选定的 k , 交换 $a[i][k]$ 和 $a[k][i]$ 。

你希望经过若干次操作以后, 这个矩阵的字典序变得尽可能小。**矩阵的字典序**是将矩阵按行的顺序展开成一个长度为 n^2 的一维数组后的字典序。换句话说, 你希望矩阵的第一行第一个元素最小; 在此基础上, 第一行第二个元素最小;; 在此基础上, 第一行第 n 个元素最小; 在此基础上, 第二行第一个元素最小;

给定初始矩阵, 请你决定进行哪些操作, 使得操作后的矩阵字典序尽可能小。输出操作后的矩阵即可。

输入格式:

第一行输入正整数 n

接下来 n 行, 每行输入 n 个用空格隔开的正整数, 表示矩阵 a

矩阵中可能有相同的数字

输出格式:

输出 n 行, 每行 n 个用空格隔开的正整数, 表示操作后的矩阵

输入样例:

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

输出样例:

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

样例解释：

操作了两次，分别选择了 $k = 2$ 和 $k = 3$ ，操作后的矩阵字典序为 $1, 2, 3, 4, 5, 6, 7, 8, 9$ 显然是最小的

数据范围：

对于30%的数据， $n \leq 10$

对于60%的数据， $n \leq 50$

对于100%的数据， $1 \leq n \leq 500, 1 \leq a_{i,j} \leq 10^9$

Problem C. 才羽绿的难题

题目英文名: midori

时空限制: 1s/256MB

题目描述:

作为千年学园游戏开发部的成员, 才羽绿和才羽桃井自然也要学习信息竞赛。

早上, 绿看到桃井正在做一道非常简单的练习题:

有一个长度为 n 的序列, 对于所有长度为 m 的子区间, 求出它的最大值。

“这不是早上刚讲过的□□□□算法(此处信息已被小钩晴删除)的模板题吗!”绿想到, “姐姐应该不会做吧。”但她却看到桃井写了一个 $O(nm)$ 的暴力: 对每个子区间从左往右暴力访问, 求出最大值:

```
for (int i = 1; i + m - 1 <= n; ++i) {
    int Max = -1;
    for (int j = i; j <= i + m - 1; ++j)
        if (a[j] > Max)
            Max = a[j]; // !
    cout << Max << endl;
}
```

绿顿时感到眼前一黑, 可见早上姐姐根本没听课, 一定不知道和谁一起跑去游戏厅了。不过绿也想到一个有趣的问题: 对于所有长度为 m 的子区间, 暴力求出最大值的时候, 最大值会被更新多少次呢, 也就是说, 上文的代码中带感叹号注释的一行会被执行多少次呢?

输入格式:

第一行输入两个正整数 n 和 m

第二行输入 n 个正整数表示序列 a

输出格式:

输出一行 $n - m + 1$ 个数, 表示每个长度为 m 的子区间会发生多少次最大值更新

输入样例:

```
7 3
1 1 4 5 1 4 4
```

输出样例：

```
2 3 2 1 2
```

样例解释：

画下划线的位置表示发生了最大值更新

第1个区间[1, 1, 4]，更新2次

第2个区间[1, 4, 5]，更新3次

第3个区间[4, 5, 1]，更新2次

第4个区间[5, 1, 4]，更新1次

第5个区间[1, 4, 4]，更新2次

数据范围：

对于30%的数据， $n \leq 10^3$

对于60%的数据，数据随机生成

对于100%的数据， $1 \leq m \leq n \leq 3 \times 10^5, 1 \leq a_i \leq 10^9$

小桃：（点提交）啊咧怎么只有30分？苦呀西！