

# N0I p 模拟考试

## 一. 题目概况

中文题目名称	括号匹配	成绩排名	字符串距离	树上染色
英文题目名称	bracket	rank	distance	color
可执行文件名	bracket	rank.cpp	distance.cpp	color.cpp
输入文件名	bracket.in	rank.in	distance.in	color.in
输出文件名	bracket.out	rank.out	distance.out	color.out
每个测试点时限	1 秒	1 秒	1 秒	2 秒
内存上限	256MB	256MB	256MB	512MB
测试点数目	10	10	20	20
每个测试点分值	10	10	5	5
结果比较方式	全文比较（过滤行末空格及文末回车）			
题目类型	传统	传统	传统	传统

## 二. 提交源程序文件名

对于 C++语言	bracket.cpp	rank.cpp	distance.cpp	color.cpp
----------	-------------	----------	--------------	-----------

## 三. 编译命令

对于 C++语言	<pre>g++ bracket.cpp -o bracket -O2 - std=c++11</pre>	<pre>g++ rank.cpp - o Rank -O2 - std=c++11</pre>	<pre>g++ distance. cpp -o distance O2 - std=c++11</pre>	<pre>g++ color.cpp -o color -O2 - std=c++11 -Wl,-- stack=268 435456</pre>
----------	---	--	---	---

## 四. 注意事项

1. 文件名（程序名和输入输出文件名）必须使用小写。
2. 选手提交以自己编号命名的文件夹，文件夹内包含 4 个源文件，并在文件夹下建立 4 个相应的子目录，并将 4 个对应的源程序分别放入对应的子文件夹中，所有名字必须使用小写；例如：

```
A01          编号
|---bracket
|    |---bracket.cpp
|---rank
|    |---rank.cpp
|---distance
|    |---distance.cpp
|---color
|    |---color.cpp
|---bracket.cpp
|---rank.cpp
|---distance.cpp
|---color.cpp
```

3. C/C++中函数 main() 的返回值类型必须是 int，程序正常结束时的返回值必须是 0，Maxlongint=2147483647。
4. 题目过于简单，请各位神犇 AK 后不要嘲讽。
5. 每道题源代码长度限制均为 50KB。
6. 评测在 Linux 系统下的全国评测机和 Windows 下的 lemon 分别评测。
7. 编译时开启 O2。
8. 建议最后 10 分钟不要再编程，检查一下提交的文件夹中的代码是否符合要求，检查文件名，输入输出文件名，数据类型，数据精度，空间限制，赋初值等是否按试卷上的要求来做的，一定要杜绝一切的不小心的人为错误，显然这种错误是致命的。
9. 做题时，审题是关键，必须深入与全面，学过的知识与做过的题都是分析问题的有利武器；编写代码要细致，多写函数，便于调试，只有这样，才能达到你的期望。

# 伢子妹子们，必须杜绝答案错误！

# 括号匹配

## 【题目描述】

给定一字符串，只由"(", ")", "\*"三种字符组成，你需要替换掉所有的"\*"号使得整个字符串变为一个平衡的括号串。你可以选择将一个"\*"替换为一个"("或")"，也可以直接删掉这个"\*"（也就是替换为一个空串）。

一个平衡的括号串定义如下：

1. 空串是平衡的括号串
2. 如果A、B都是平衡的括号串，那么AB也是一个平衡的括号串（这里AB指将A与B拼接起来）
3. 如果A是一个平衡的括号串，那么(A)也是一个平衡的括号串

给定这样一个含有"\*"的括号串，输出替换掉所有"\*"后得到的一个平衡的括号串。由于可能存在多种答案，所以本题只要求选手输出长度最短且字典序最小的括号串（即所有长度最短的答案中字典序最小的一个）。如果不能得到一个平衡的括号串，输出"No solution!"。

给定两个长度均为 $n$ 的字符串 $A$ 和 $B$ ，如果存在一个整数 $k$ 满足 $1 \leq k \leq n$ ， $A$ 和 $B$ 的前 $k-1$ 个字符完全相等，且 $A$ 的第 $k$ 个字符小于 $B$ ，我们就称 $A$ 的字典序小于 $B$ 。例如，" $()$ "的字典序小于" $()()$ "的字典序。

## 【输入】

输入仅一行，即一个字符串，只含有"(", ")", "\*"三种字符。

## 【输出】

输出一行，即替换掉所有"\*"后长度最短且字典序最小的平衡括号串。如果无解，输出"No solution!"。

## 【样例输入】

```
((***)()((**
```

## 【样例输出】

```
((()))()((()))
```

## 【数据范围】

设字符串长度为 $n$ ：

对于20%的数据，有 $1 \leq n \leq 15$

对于40%的数据，有 $1 \leq n \leq 10^3$

对于100%的数据，有 $1 \leq n \leq 10^5$

# 成绩排名

## 【题目描述】

小Z毕业后去了H中学教书，他带的班级有 $n$ 个学生，对于每个学生 $i$ 可以用一个正整数 $A_i$ 来衡量其学习能力（ $i = 1, 2, \dots, n$ ）。有一天，小Z捡到了 $k$ 本神奇教材，如果给一个学生学习了这本教材，他的学习能力就会变成原来的 $t$ 倍。小Z决定将这 $k$ 本书随机且等概率地发放给班上的 $n$ 个同学，每个学生最多只能得到一本书，也就是说班上会有 $k$ 名同学被分配到书，学习能力变成 $k$ 倍，而剩下 $n - k$ 名同学没有书，学习能力不变。现在小Z想知道，在所有的分配方案中，对于班上的每个同学来说排名保持不变的情况分别有多少种。（在这里，排名是指学习能力比他高的人的数目加一，比方说有5个同学，学习能力为[3,2,2,1,1]，他们的排名即[1,2,2,4,4]。）

## 【输入】

第一行输入3个整数 $n, k, t$ ，分别代表学生的数目、神奇教材的数目以及学习能力翻的倍数。第二行输入 $n$ 个整数 $A_i$ ，代表每个学生的学习能力，编号从1到 $n$ 。

## 【输出】

输出 $n$ 行，第 $i$ 行输出一个整数代表第 $i$ 位同学在Z老师发完书后学习能力排名保持不变的所有可能的情况数目。由于这些数可能非常大，请输出其对 $10^9 + 7$ 取模后的结果。

## 【样例输入】

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

## 【样例输出】

```
4
3
2
4
```

## 【数据范围】

- 对于10%的数据，有 $1 \leq n \leq 10$
- 对于30%的数据，有 $1 \leq n \leq 10^3$
- 对于另外10%的数据，有 $1 \leq n \leq 10^5$ 且 $k = 1$
- 对于100%的数据，有 $1 \leq n \leq 10^5, 1 \leq k \leq n, 2 \leq t \leq 1000, 1 \leq A_i \leq 10^9$

# 字符串距离

## 【题目描述】

对于两个小写字母组成的字符串 $S$ 和 $T$ ，规定一种操作，每次操作可以选择 $S$ 或 $T$ 上任意一个字符并删除这个字符，或选择 $S$ 或 $T$ 上任意一个位置，在这个位置上插入任意一个小写字母（也可以选择字符串的开头或结尾）。你可以通过执行若干次上述操作，来使得两个字符串变得完全相同，规定两字符串之间的距离为使得两字符串变得完全相同所需要的最少操作次数。

现给定两个字符串 $A$ 和 $B$ ，以及 $q$ 次询问，每次询问以两个整数 $l$ 和 $r$ 的形式给出，你需要求出字符串 $A[l \dots r]$ （即字符串 $A$ 的区间 $[l, r]$ 上的字符组成的子串）和字符串 $B$ 的距离是多少。

## 【输入】

第一行输入一个字符串 $A$ ，第二行输入一个字符串 $B$ ，第三行输入一个正整数 $q$ ，表示询问的次数。

接下来 $q$ 行，每次输入两个整数 $l$ 和 $r$ ，代表区间 $[l, r]$ 。

## 【输出】

对于每次询问，输出一个正整数，表示字符串 $A[l \dots r]$ 和字符串 $B$ 的距离。

## 【样例输入】

```
qaqaqwqaqaq
qaqwqaq
3
1 7
2 8
3 9
```

## 【样例输出】

```
4
2
0
```

## 【数据范围】

对于40%的数据，有 $1 \leq n \leq 500$ ， $1 \leq q \leq 10^3$

对于100%的数据，有 $1 \leq n \leq 10^5$ ， $1 \leq q \leq 10^5$

# 树上染色

## 【题目描述】

现有一颗含有 $n$ 个结点的有根树，根固定为标号为1的结点。对于树上每个结点 $i$  ( $i = 1, 2, \dots, n$ )都有一个权值 $val_i$ 和一种颜色 $col_i$ ，颜色标号从1到 $n$ ，代表 $n$ 种颜色。

现有两种操作如下：

- $1 \times v$  将结点 $x$ 的权值设为 $v$
- $2 \times c$  将结点 $x$ 的颜色设为 $c$

对于所有满足 $1 \leq u < v \leq n$ 且不构成祖先关系（即 $u$ 不是 $v$ 的祖先结点，且 $v$ 不是 $u$ 的祖先结点）的一对结点 $u, v$ ，规定 $f(u, v)$ 的值如下，其中 $\oplus$ 是按位异或符号：

$$f(u, v) = \begin{cases} val_u \oplus val_v, & col_u = col_v \\ 0, & col_u \neq col_v \end{cases}$$

给定 $q$ 个操作，要求选手求出对于第 $i$ 次 ( $i = 0, 1, 2, \dots, n$ ) 操作后 $S$ 的值（第0次的结果即在所有操作开始前 $S$ 的值），其中 $S$ 的值规定为

$$\sum_{\substack{1 \leq u < v \leq n \\ u \text{ 和 } v \text{ 不构成祖先关系}}} f(u, v)$$

## 【输入】

第一行输入一个正整数 $n$ 表示结点个数。

第二行输入 $n$ 个正整数，表示结点 $i$ 的初始颜色 $col_i$ 。

第三行输入 $n$ 个正整数，表示结点 $i$ 的初始权值 $val_i$ 。

接下来的 $n - 1$ 行，每行输入两个正整数 $u$ 和 $v$  ( $1 \leq u, v \leq n$ )，表示树上一条连接了结点 $u$ 和结点 $v$ 的边。题目保证所给数据可以组成一棵合法的树。

接下来的一行输入一个正整数 $q$ 表示操作的次数。

接下来的 $q$ 行，每行输入3个整数，如果第一个整数的值为1，则代表第一种操作，那么后两个数 $x$ 和 $v$ 就表示将结点 $x$ 的权值设为 $v$ ；如果第一个整数的值为2，则代表第二种操作，那么后两个数 $x$ 和 $c$ 就表示将结点 $x$ 的颜色设为 $c$ 。

请选手注意：数据量较大，请尽量使用快速读入方式来读入数据。

## 【输出】

输出 $q + 1$ 行，每行表示第 $i$ 次 ( $i = 0, 1, 2, \dots, n$ ) 操作后 $S$ 的值。

## 【样例输入】

```
5
1 1 1 1 1
1 2 4 8 16
1 2
3 1
2 4
2 5
2
1 3 32
2 3 2
```

### 【样例输出】

```
62
146
24
```

### 【数据范围】

对于前10%的数据，有 $1 \leq n \leq 100$

对于前20%的数据，有 $1 \leq n \leq 1000, 1 \leq q \leq 100$

对于前30%的数据，有 $1 \leq n \leq 1000, 1 \leq q \leq 1000$

对于前60%的数据，有 $1 \leq n \leq 10^5, 1 \leq q \leq 20000, 0 \leq val_i < 2^{15}$

对于全部的数据，有 $1 \leq n \leq 10^5, 1 \leq q \leq 10^5, 1 \leq col_i \leq n, 0 \leq val_i < 2^{20}$