

3031_字符出现次数

描述

编写一个程序，输入一个字符串，然后统计该字符串中，每一个字母的出现次数。

输入

输入一个字符串（长度不超过 128 个字符）。将其中的大写字母转换为小写字母，并忽略 26 个字母之外的任意字符。

输出

输出只有一行，包括 26 个整数，即每个字母的出现次数。

输入样例

```
Friends, Romans, countrymen, lend my you ears;
```

输出样例

```
2 0 1 2 4 1 0 0 1 0 0 1 3 5 3 0 0 4 3 1 2 0 0 0 3 0
```

3026_字符串进制转换

描述

编写一个程序，输入一个二进制的字符串（长度不超过 32），然后计算出相应的十进制整数，并把它打印出来。

输入

输入一个二进制字符串。

输出

输出相应的十进制整数。

输入样例

```
1101
```

输出样例

```
13
```

2016_质数之和

描述

编写一个程序，由用户输入一个整数 N ，然后计算出前 N 个质数的和。例如：如果 $N=3$ ，那么结果为 $2+3+5=10$ ；如果 $N=7$ ，那么结果为 $2+3+5+7+11+13+17=58$ 。

输入

输入一个正整数 N 。

输出

输出一个整数，即前 N 个质数之和

输入样例

3

输出样例

10

020_Fibonacsl 数列

描述

计算 Fibonacsl 数列中的第 n 个元素的值。所谓 Fibonacsl 数列，即满足下列公式的整数序列：

$$F_n = a * F_{n-2} + b * F_{n-1}$$

其中， a 、 b 是任意给定的整数常量，序列的初始值 F_1 和 F_2 也是任意给定的。例如，如果 $F_1=3$ 、 $F_2=1$ 、 $a=2$ 、 $b=-3$ ，则相应的 Fibonacsl 数列为：

3, 1, 3, -7, 27, -95, 339, ...

请编写一个程序，对于用于的参数，计算相应数列中第 n 个元素的值。

输入

输入只有一行，包括 5 个整数： F_1 、 F_2 、 a 、 b 和 n ，其中 $n < 100$ 。

输出

输出相应数列的第 n 个元素。

输入样例

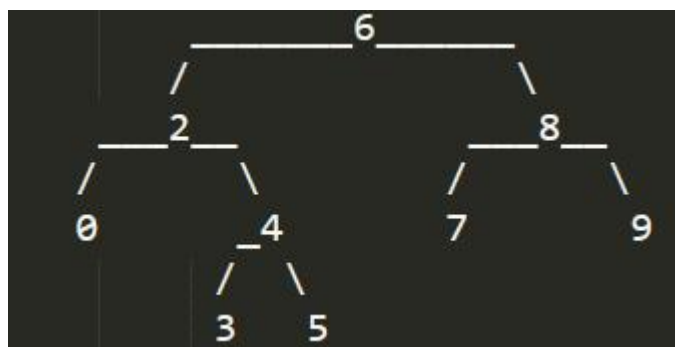
3 1 2 -3 5

输出样例

27

二叉搜索树的最小公共祖先

给定一个二叉搜索树，对于其中任意两个节点，我们都可以发现其最小公共祖先。如下面这棵树



0 和 9 的公共祖先为 6，3 和 5 的公共祖先为 4。元素本身可以是其自己的祖先。所以 4 和 5 的祖先是 4，6 和 6 的祖先为 6。

事实上，给定一个输入序列，在不做任何平衡操作的情况下，我们可以通过不断在对这颗二叉搜索树做插入操作，从而唯一的确定这颗二叉搜索树。如这棵树就可以由输入序列：**6 2 8 0 4 7 9 3 5** 确定。当然，这样的输入序列并不是唯一的。

输入

第一行两个数 **N M**。其中 **N** 代表数据的个数，**M** 代表样本数。 ($N \leq 50000$, $M \leq 1000$, 数据随机产生)

接下来一行有 **N** 个数，代表输入序列，用于构造这个二叉搜索树。

接下来 **M** 行，每行两个数，代表待查询的数。

输出

M 行，每行 1 个数，代表待查询数的最小公共祖先。

样例输入

```
9 5
6 2 8 0 4 7 9 3 5
0 9
3 5
5 4
6 6
```

0 5

样例输出

6

4

4

6

2