# 二叉树(binary)

时间限制: 1 Sec 内存限制: 128 MB

### 题目描述

在一棵无穷大的二叉树中:

- 1.每个节点都有左右两个儿子
- 2.如果一个节点的标号为x,则它的左儿子标号为2\*x,右儿子标号为2\*x+1
- 3.根节点的标号为1。

从根开始行走,每一步要么跳到左儿子,或者跳到右儿子,或者停在原节点。

- 一次行走,可以用一个包含'L','R','P'三种字符的字符串表示。
- 一次行走的值,是最后停下位置的标号值。例如"LR"的值为5,"RPP"的值为3。
- '\*'可以表示这三个字符中的任何一个。

给定一个字符串,输出所有可能值的总和。

#### 输入

输入一行,只包含'L','R','P','\*'的字符串。

### 输出

输出一个整数,表示所有可能值的和。

## 样例输入

L\*R

\*\*

LLLLLRRRRRRLLL

### 样例输出

25		
33		
8440		

## 提示

第1~2组测试数据,字符串长度不超过20,没有'\*' 第3组测试数据,字符串长度不超过200,没有'\*' 第4~6组测试数据,字符长度不超过1000,'\*'不超过5个 第7~10组测试数据,字符长度不超过10000

# 字符串转换(string)

时间限制: 1 Sec 内存限制: 128 MB

### 题目描述

佳佳有两个字符串s, w,他可以进行下列三种操作:

从第一个字符串中删去任意一个字符,需要a分钟;

在第一个字符串的任意一个位置插入一个字符,需要a分钟;

将第一个字符串中某位置上的字符替换成另一个字符,需要b分钟。

佳佳一共有k分钟来进行操作,请你告诉佳佳最少需要多少分钟将s转换成w

注意 , |s|不一定等于|w|

## 输入

第一行包括一个字符串8。

第二行包括一个字符串 w。

第三行包含三个整数a, b, k。

### 输出

输出最少时间,如果不能再k分钟之内完成,输出-1

## 样例输入

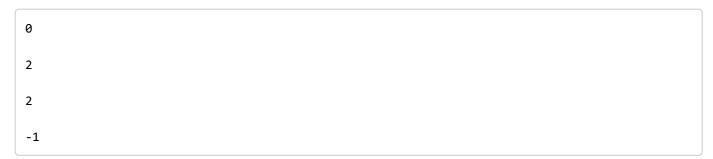
```
aaa
bbbb
0 0 100

abab
acac
1 1 100

baaaaa
aaaaab
1 100 100

aaaaaa
bbbbbb
100 100 0
```

## 样例输出



# 提示

```
对于20\%的数据,1\leq |s|,|w|\leq 400.
对于60\%的数据,1\leq |s|,|w|\leq 4000.
对于100\%的数据,1\leq |s|,|w|\leq 10^5,a,b,k\leq 100
```

## 公交路线(bus)

时间限制: 2 Sec 内存限制: 256 MB

### 题目描述

到了周末,C校的同学都决定出去玩!

一共有M位同学,每位同学都有一个想到达的目的地,勤俭节约的他们都决定乘坐公交车到达.

C校的公交车一共有T条线路,N个站点.N个站点之间会构成一棵树.

现在给出站点的分布,以及T条路线的起点和终点,对于每个学生求出他回到家需要换多少次车才能回到家

如果无法到家,输出-1.

### 输入

第一行3个整数N,T,M,分别表示站点数,公交路线数和同学的数量,站点的编号为 $1,2,3,\cdots$ ,N。

接下来N-1行,每行两个正整数x,y表示x和站点y之间有一条道路.

接下来T行,每行两个正整数s,t,表示有一条公交路线的起点站为s,终点站为t,线路是双向的。 $(s \neq t)$ 

最后M行,每行同样有两个正整数a,b,表示该同学想从站点a到站点b.(  $a \neq b$ )

### 输出

-共M行,每行-个整数.

第1行的整数表示第1个同学从起点到达目的地最少需要换乘的次数.

如果不能到达,输出-1.

### 样例输入

```
      7 3 2

      1 2

      1 3

      2 4

      2 5

      3 6

      3 7

      3 1

      4 5

      6 7

      1 6

      4 7
```

## 样例输出

```
1
-1
```

## 提示

如果把公交线路按输入的顺序依次编号,从1到6,只需先乘1路车从1到3,然后换乘3路从3到6即可,而从4到7却无法仅用公交车,因为没有任何一路车覆盖2-1这条路。

测试点编号	N, M, T		
1~4	≤100	无	
5 <sup>~</sup> 8	≤1000		
9 <sup>~</sup> 12		保证站点构成一条链	
13 <sup>~</sup> 16	≤100000	以1为根,每条公交路线的起点和终点一定是祖孙关系	
17 <sup>~</sup> 20		无	