Overview

Author	Timothy Stranex	Bruce Merry	BIO 2002
Problem	**york**	**joke**	**mouse**
Source	york.java york.py york.c york.cpp york.pas	joke.java joke.py joke.c joke.cpp joke.pas	mouse.java mouse.py mouse.c mouse.cpp mouse.pas
Input file	york.in	joke.in	mouse.in
Output file	york.out	joke.out	mouse.out
Time limit	1 second	1 second	1 second
Number of tests	10	10	10
Points per test	10	10	10
Total points	**100**	**100**	**100**

The maximum total score is 300 points.

Four Yorkshiremen

Author

Timothy Stranex

Introduction

The Four Yorkshiremen have got onto the topic of how many prime palindromes they had to learn as children.

Yorkshireman 1: "And when I got back from the Mill, I had to recite the first thirty prime palindromes or get no supper."

Yorkshireman 2: "Only the thirty? I had to recite the first fifty just to be allowed to go work at the Mill in the morning."

Yorkshireman 3: "I used to dream of only having to memorise the first fifty."

Yorkshireman 4: "That's nothing. I had to know all the prime palindromes less than forty thousand. And if I got one wrong, I had to go without food for a week."

The Yorkshiremen are getting old and they can't work out quickly how many prime palindromes are less than a certain number. This is important since they need to be able to gauge the others' statements to keep competing.

Task

You must help the Yorkshiremen by writing a program to count all the prime palindromes less than some integer N.

An integer is prime if it is greater than or equal to two and has no other factors besides 1 and itself. For example, 12 has factors 1, 2, 3, 4, 6 and 12 so it is not prime but 5 is prime because its only factors are 1 and 5.

An integer is a palindrome if it is the same when its decimal representation is read forwards or backwards. For example, 13531 and 1221 are palindromes but 12 and 596 are not.

Example

Suppose $N = 17$. The prime numbers less than seventeen are 2, 3, 5, 7, 11 and 13. Of these, all are palindromes except 13. Therefore, there are five prime palindromes less than seventeen.

Input (york.in)

The input consists of a single line containing the integer N.

Sample input

```
17
```

Output (york.out)

The output consists of a single line containing the number of prime palindromes less than N.

Sample output

```
5
```

Constraints

- $1 \leq N \leq 80000$

In 50% of the test cases, $1 \leq N \leq 5000$.

Time limit

1 second. Python: 10 seconds.

Funniest Joke

Author

Bruce Merry

Introduction

Ernest Scribbler has written the funniest joke in the world. The joke is so funny that anyone who reads it dies almost instantly. The British army soon noticed its potential as a weapon and translated it into German to use against the Nazis. They will transmit the joke to their ground troops in an encrypted form to prevent it from being read accidentally and to keep it secret from Nazi spies. Help them check whether their encryption algorithm is reasonable by writing a program to perform letter frequency counts on the encrypted messages.

Task

The algorithm is to take each character of the original message and replace it with some string of characters (depending on the original character). Since keeping the joke a secret is so important, they may do this multiple times.

Your program will be given an original message, and a character range. It must return the frequency of each letter in the given range of the encrypted string.

Example

Suppose the original message is “badbeef”. The replacement table is as follows:

$$a \to dc \qquad b \to aa$$
$$c \to facb \qquad d \to a$$
$$e \to ea \qquad f \to f$$

After one iteration, the message would be “aadcaaaeaeaf”. After a second iteration it becomes “dcdcafacbdcdcdceadceadcf”. The frequencies of letters between positions 4 and 15 (underlined) are 2 a’s, 1 b, 5 c’s, 3 d’s and 1 f.

Input (joke.in)

The first line contains four integers separated by spaces, L, A, B and N. Only the first L letters of the alphabet are used in both the original messages and the replacement table. You are asked for the frequencies between positions A and B in the encryption (inclusive, and counting from 1). N is the number of iterations of the encryption algorithm.

The next L lines each contain a non-empty string of lower-case letters. The first is the replacement string for “a”, the second for “b” and so on. The following line contains another non-empty string of lower-case letters, which is the message that is going to be encrypted.

Sample input

```
6 4 15 2
dc
aa
facb
a
ea
f
badbeef
```

Output (joke.out)

The output consists of L integers, one per line. The first is the number of a’s in the range, the second the number of b’s and so on.

Sample output

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

Constraints

- $1 \leq A \leq B < 2^{30}$
- $1 \leq L \leq 26$
- $1 \leq N \leq 20$
- $1 \leq$ length of each replacement string ≤ 4
- $1 \leq$ length of the original message ≤ 10

In 50% of the test cases, $B - A \leq 10^6$.

Time limit

1 second. Python: 10 seconds.

The Mouse Problem

Author

BIO 2002

Introduction

The police have started to infiltrate the mouse parties, dressing up as voles so that they can identify closet mice. They have been quite successful so far because few of the mice know each other, and so the infiltrators do not obviously stand out. Help the mice to get to know each other better, so that they cannot be so easily infiltrated.

Task

The mice are going to stand in a circle and eat cheese. The goal is to have each mouse stand between two other mice that he/she does not know, to whom he/she can then be introduced. Fortunately this will be relatively easy to arrange, because each mouse knows less than half of the other mice.

Example

Suppose there are 6 mice, numbered 1 to 6. The pairs (1, 2), (2, 3), (3, 4), (4, 5), (5, 6) and (6, 1) all know each other (it is assumed that if A knows B then B knows A). If the mice stand in the order 1, 4, 2, 6, 3, 5 around the circle, then no pair of neighbouring mice will know each other.

Input (mouse.in)

The first line of input contains the integers N and F, separated by a space. N is the number of mice (numbered from 1 to N) and F is the number of pairs that know each other. The following F lines each contain two integers A and B separated by a space, indicating that mice A and B know each other. It is guaranteed that $A \neq B$ and that no pair is listed twice.

Sample input

```
6 6
1 2
2 3
3 4
4 5
5 6
6 1
```

Output (mouse.out)

The output consists of N lines, each of which is the number of a mouse. The numbers indicate the order in which the mice stand in the circle.

Sample output

```
1
4
2
6
3
5
```

Constraints

- $3 \leq N \leq 1000$

In 50% of the test cases, $N \leq 20$.

Time limit

1 second. Python: 10 seconds.