A. Grammar Lessons

time limit per test: 5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Petya got interested in grammar on his third year in school. He invented his own language called Petya's. Petya wanted to create a maximally simple language that would be enough to chat with friends, that's why all the language's grammar can be described with the following set of rules:

- There are three parts of speech: the adjective, the noun, the verb. Each word in his language is an adjective, noun or verb.
- There are two genders: masculine and feminine. Each word in his language has gender either masculine or feminine.
- Masculine adjectives end with `-lios`, and feminine adjectives end with `-liala`.
- Masculine nouns end with `-etr`, and feminime nouns end with `-etra`.
- Masculine verbs end with `-initis`, and feminime verbs end with `-inites`.
- Thus, each word in the Petya's language has one of the six endings, given above. There are no other endings in Petya's language.
- It is accepted that the whole word consists of an ending. That is, words "`lios`", "`liala`", "`etr`" and so on belong to the Petya's language.
- There aren't any punctuation marks, grammatical tenses, singular/plural forms or other language complications.
- A sentence is either exactly one valid language word or exactly one *statement*.

Statement is any sequence of the Petya's language, that satisfy both conditions:

- Words in statement follow in the following order (from the left to the right): zero or more adjectives followed by exactly one noun followed by zero or more verbs.
- All words in the statement should have the same gender.

After Petya's friend Vasya wrote instant messenger (an instant messaging program) that supported the Petya's language, Petya wanted to add spelling and grammar checking to the program. As Vasya was in the country and Petya didn't feel like waiting, he asked you to help him with this problem. Your task is to define by a given sequence of words, whether it is true that the given text represents exactly one sentence in Petya's language.

Input

The first line contains one or more words consisting of lowercase Latin letters. The overall number of characters (including letters and spaces) does not exceed 10^5.

It is guaranteed that any two consecutive words are separated by exactly one space and the input data do not contain any other spaces. It is possible that given words do not belong to the Petya's language.

Output

If some word of the given text does not belong to the Petya's language or if the text contains more that one sentence, print "`NO`" (without the quotes). Otherwise, print "YES" (without the quotes).

Examples

input
```
petr
```
output
```
YES
```

input
```
etis atis animatis etis atis amatis
```
output
```
NO
```

input
```
nataliala kataliala vetra feinites
```
output
```
YES
```

B. Petr#

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Long ago, when Petya was a schoolboy, he was very much interested in the Petr# language grammar. During one lesson Petya got interested in the following question: how many different continuous substrings starting with the s_{begin} and ending with the s_{end} (it is possible $s_{begin} = s_{end}$), the given string t has. Substrings are different if and only if their contents aren't equal, their positions of occurence don't matter. Petya wasn't quite good at math, that's why he couldn't count this number. Help him!

Input

The input file consists of three lines. The first line contains string t. The second and the third lines contain the s_{begin} and s_{end} identificators, correspondingly. All three lines are non-empty strings consisting of lowercase Latin letters. The length of each string doesn't exceed 2000 characters.

Output

Output the only number — the amount of different substrings of t that start with s_{begin} and end with s_{end}.

Examples

input

```
round
ro
ou
```

output

```
1
```

input

```
codeforces
code
forca
```

output

```
0
```

input

```
abababab
a
b
```

output

```
4
```

input

```
aba
ab
ba
```

output

```
1
```

Note

In the third sample there are four appropriate different substrings. They are: ab, abab, ababab, abababab.

In the fourth sample identificators intersect.

C. Double Happiness

time limit per test: 3 seconds
memory limit per test: 128 megabytes
input: standard input
output: standard output

On the math lesson a teacher asked each pupil to come up with his own lucky numbers. As a fan of number theory Peter chose prime numbers. Bob was more original. He said that number t is his lucky number, if it can be represented as:

$$t = a^2 + b^2,$$

where a, b are arbitrary positive integers.

Now, the boys decided to find out how many days of the interval $[l, r]$ ($l \le r$) are suitable for pair programming. They decided that the day i ($l \le i \le r$) is suitable for pair programming if and only if the number i is lucky for Peter and lucky for Bob at the same time. Help the boys to find the number of such days.

Input

The first line of the input contains integer numbers l, r ($1 \le l, r \le 3 \cdot 10^8$).

Output

In the only line print the number of days on the segment $[l, r]$, which are lucky for Peter and Bob at the same time.

Examples

input

```
3 5
```

output

```
1
```

input

```
6 66
```

output

```
7
```

D. Museum

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

One day as Petya and his friend Vasya were having one of their numerous trips, they decided to visit a museum castle. The museum has a specific shape: it consists of n rooms connected with m corridors so that one can access any room from any other one.

After the two friends had a little walk around the museum, they decided to split and watch the pieces of art each of them found interesting. They agreed to meet in one of the rooms at six p.m. However, they forgot one quite essential thing: they didn't specify the place to meet and when the time came, they started to rush about the museum looking for each other (they couldn't call each other as roaming made a call's cost skyrocket).

Yet, even despite the whole rush, they couldn't get enough of the pieces of art, that's why each of them has the following strategy: each minute he make a decision where to go — with probability p_i he doesn't move to any other place during this minute (i.e. he stays in the room). With probability $1 - p_i$ he equiprobably choose one of the adjacent rooms and went there along the corridor. Here i is the ordinal number of the current room. Building was expensive in ancient times, that's why each corridor connected two different rooms, and any two rooms had no more than one corridor between them.

The boys act simultaneously. As the corridors are dark, it is impossible to meet there; however, one can walk along the corridors in both directions (besides, the two boys can be going through the same corridor simultaneously without meeting). The boys act like that until they meet each other. More formally, the two friends meet when at some moment of time both of them decided to appear in the same room.

For each room find the probability that the boys will meet there considering that at 6 p.m. they are positioned in rooms a and b correspondingly.

Input

The first line contains four integers: n $(1 \le n \le 22)$, representing the numbers of rooms; m $(n - 1 \le m \le \frac{n(n-1)}{2})$, representing the number of corridors; a, b $(1 \le a, b \le n)$, representing the numbers of Petya's and Vasya's starting rooms correspondingly.

Next m lines contain pairs of numbers — the numbers of rooms connected by a corridor. Next n lines contain probabilities p_i $(0.01 \le p_i \le 0.99)$ with the accuracy of up to four digits after the decimal point — the probability to stay in room i.

It is guaranteed that every room can be reached from every other room by corridors.

Output

In the only line print n space-separated numbers, the i-th number should represent the probability that the friends meet in the i-th room with absolute or relative error of no more than 10^{-6}.

Examples

input

```
2 1 1 2
1 2
0.5
0.5
```

output

```
0.5000000000 0.5000000000
```

input

```
4 4 1 2
1 2
2 3
3 4
4 1
0.5
0.5
0.5
0.5
```

output

```
0.3333333333 0.3333333333 0.1666666667 0.1666666667
```

Note

In the first sample the museum is symmetric. That means the probabilities to meet in rooms 1 and 2 are equal. And their sum equals to one. So, each probability equals 0.5.

E. Sleeping

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

One day Vasya was lying in bed watching his electronic clock to fall asleep quicker.

Vasya lives in a strange country, where days have h hours, and every hour has m minutes. Clock shows time in decimal number system, in format `H:M`, where the string `H` always has a fixed length equal to the number of digits in the decimal representation of number $h - 1$. To achieve this, leading zeros are added if necessary. The string `M` has a similar format, and its length is always equal to the number of digits in the decimal representation of number $m - 1$. For example, if $h = 17$, $m = 1000$, then time equal to 13 hours and 75 minutes will be displayed as "`13:075`".

Vasya had been watching the clock from h_1 hours m_1 minutes to h_2 hours m_2 minutes inclusive, and then he fell asleep. Now he asks you to count how many times he saw the moment at which at least k digits changed on the clock simultaneously.

For example, when switching 04:19 → 04:20 two digits change. When switching 23:59 → 00:00, four digits change.

Consider that Vasya has been watching the clock for strictly less than one day. Note that the last time Vasya saw on the clock before falling asleep was "`h2:m2`". That is, Vasya **didn't see** the moment at which time "`h2:m2`" switched to the next value.

Input

The first line of the input file contains three space-separated integers h, m and k ($2 \le h, m \le 10^9$, $1 \le k \le 20$). The second line contains space-separated integers h_1, m_1 ($0 \le h_1 < h$, $0 \le m_1 < m$). The third line contains space-separated integers h_2, m_2 ($0 \le h_2 < h$, $0 \le m_2 < m$).

Output

Print a single number — the number of times Vasya saw the moment of changing at least k digits simultaneously.

Please do not use the `%lld` specificator to read or write 64-bit integers in C++. It is preferred to use the `cin` stream (also you may use the `%I64d` specificator).

Examples

input

```
5 5 2
4 4
2 1
```

output

```
3
```

input

```
24 60 1
0 0
23 59
```

output

```
1439
```

input

```
24 60 3
23 59
23 59
```

output

```
0
```

Note

In the first example Vasya will see the following moments of time: 4:4 ⇒ 0:0 → 0:1 → 0:2 → 0:3 → 0:4 ⇒ 1:0 → 1:1 → 1:2 → 1:3 → 1:4 ⇒ 2:0 → 2:1 → 2:2 → 2:3 → 2:4. Double arrow (⇒) marks the sought moments of time (in this example — when Vasya sees two numbers changing simultaneously).

In the second example $k = 1$. Any switching time can be accepted, since during switching of the clock at least one digit is changed. Total switching equals to $24 \cdot 60 = 1440$, but Vasya have not seen one of them — the switching of 23:59 ⇒ 00:00.

In the third example Vasya fell asleep immediately after he began to look at the clock, so he did not see any change.