

A. Black moves first and...

You are given the positions of the white king, a white rook, and the black king on a chessboard. Each position is given by two characters: a letter from “a” to “h” specifying the column and a digit from 1 to 8 specifying the row. In the given position, black player moves first.

Recall that a rook attacks all squares in the same row or column, except when there is another chess piece between the rook and the respective square. A king attacks all squares adjacent to its position by side or corner, and can move (either taking a piece of the opposite color or not) to any of these squares given that it is not attacked by a piece of the opposite color.

You have to classify the given position as one of the following classes:

- strange: the black king is attacked by the white king,
- normal: the black king is not attacked and can make a valid move,
- check: the black king is attacked and can make a valid move,
- stalemate: the black king is not attacked and can not make a valid move,
- checkmate: the black king is attacked and can not make a valid move.

Input format

The first line contains three pairs of characters. The first character of each pair is a letter from “a” th “h”, and the second one is a digit from 1 to 8. These three pairs specify the positions of the white king, a white rook and the black king respectively. Consecutive pairs of characters are separated by spaces. It is guaranteed that no two pieces occupy the same square.

Output format

If the position is strange, print “Strange”. If the position is normal, print “Normal”. If the position is a check, print “Check”. If the position is a stalemate, print “Stalemate”. If the position is a checkmate, print “Checkmate”.

Sample 1

Input

b2 a3 c4

a1 b2 e2

Output

Normal

Check

B. Byteland Shamans

The shamans who lived in Byteland in ancient times used the following way to construct spells.

First, two base phrases A and B are selected. Each phrase can consist of several words. After that, the following sequence of spells was constructed: $T_0 = A$, $T_1 = B$, $T_2 = A B$, $T_3 = B A B$, ..., $T_n = T_{n-2} T_{n-1}$.

As you can see, each spell T_i starting from $i = 2$ consisted of performing spells T_{i-2} and T_{i-1} in sequence. While performing spells in sequence, shamans included a small pause between them, so no new words could occur as a result of concatenation.

Different causes required different spells: for example, for the rite of initiation, a shaman should perform such a spell that the name of spirit S will be pronounced exactly K times. The name S occurs in at least one of the base phrases. Anthropologists made a request for you: knowing the base phrases, the name of spirit S which occurs in at least one of them and the number K , determine N : the minimum number of the spell T_N such that S will be pronounced exactly K times while performing that spell.

Input format

The first line contains the base phrase A : a string of no more than 255 ASCII characters with codes from 32 to 127 inclusive. The phrase contains from one to ten words.

A word is a sequence of English letters bounded by separator characters which are all non-letter characters. The upper-case and lower-case letters are considered the same.

The second line contains the base phrase B in the same format.

The third line contains the name of spirit S : a word consisting of lower-case English letters. It is guaranteed that S is contained in at least one of the base phrases A and B .

The fourth line contains a positive integer K . The length of this line does not exceed 3000 characters.

Output format

Print one integer N : the minimum number of the spell T_N such that S will be pronounced exactly K times while performing that spell. If there is no such number, print "Impossible" instead.

Sample 1

Input

```
Python is great!  
Python-Die-Grosse-Shlange  
python  
5
```

Output

```
4
```

Sample 2

Input

Great Python, answer to me!

Help us, Python

python

4

Output

Impossible

C. Circles-2

You are given n triangles with integer lengths of sides. Calculate the maximum possible number of triangles which have the same radius of the incircle.

Input format

The first line contains one integer n ($1 \leq n \leq 1000$). Each of the next n lines contains three integers a_i, b_i, c_i : the sides of i -th triangle. It is guaranteed that the triangle is non-degenerate and $1 \leq a_i, b_i, c_i \leq 1000$.

Output format

Print a single integer: the maximum number of triangles with the same radius of the incircle.

Sample

Input

```
3
3 4 5
5 4 3
10 10 10
```

Output

```
2
```

D. After the Finals

Byteland Championship Finals are held in city $\mathcal{I}$ for the first time. The road network of city $\mathcal{I}$ consists of N crossings connected by M bidirectional roads. The schedule of the finals was so tight that the participants did not have time to visit the city. In addition, the closing ceremony took longer than expected, so when the winning team departed, they had to pick the shortest route to the airport. Still, they wanted to see a bit of the city nevertheless, so of all possible shortest routes, they picked one that contained the maximum possible number of roads.

Given the map of the city, calculate the length of the route and the number of roads visited by the team.

Input format

The first line contains two integers N and M : the number of crossings and roads of city $\mathcal{I}$ respectively ($2 \leq N \leq 1000$, $1 \leq M \leq 2 \cdot 10^5$).

Each of the next M lines contains three integers a_i , b_i and c_i ($1 \leq a_i, b_i \leq N$, $a_i \neq b_i$, $1 \leq c_i \leq 1000$): the numbers of crossings connected by i -th road and the length of that road. Two different crossings may be connected by more than one road. The finals took place at crossing number 1 , and the airport is at crossing number N .

It is guaranteed that there is at least one route from where the finals took place to the airport.

Output format

Print two integers P and Q : the length of the shortest route from where the finals took place to the airport, and the maximum number of roads in a shortest route between these places.

Sample

Input

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

Output

```
2 2
```

E. Factorials

Misha recently learned that a factorial of an integer n is the product of all positive integers from 1 to n inclusive. A factorial of n is denoted as $n!$; so, for example, $1! = 1$, $2! = 1 \cdot 2 = 2$, $3! = 1 \cdot 2 \cdot 3 = 6$. By definition, $0! = 1$.

Misha liked to calculate the factorials of small integers. However, he quickly understood that the factorials of numbers greater than ten are hard to calculate by hand. So he turned to calculating only a few last digits, but soon found out they are often zeroes: for example, if his calculations were right, the last six digits of the number $25!$ are zeroes. The problem of finding the last non-zero digit of $n!$ proved very interesting, but Misha finally solved that one, too. Misha's mathematics teacher was pleased by Misha's success and gave him a harder problem: find the last non-zero digit of the sum of k -th powers of factorials of integers from 0 to n . Here, k is a non-negative integer. Because Misha is a sixth grade student, the teacher told him to solve the problem only for $k \leq 3$ and $n \leq 100$. Still, this problem involves heavy calculations if done by hand...

Help Misha to write a computer program in order to solve the given problem. Write a program which, given n and k , will calculate the last non-zero digit of the sum of k -th powers of factorials of all integers from 0 to n inclusive.

Input format

The first line contains two integers: a positive integer n which does not exceed 100 and a non-negative integer k which is at most three.

Output format

Print one integer: the last non-zero digit of the sum of k -th powers of factorials of all integers from 0 to n inclusive.

Sample 1

Input

4 1

Output

4

Sample 2

Input

3 2

Output

2

F. Opposite Continued Fraction

A *finite continued fraction* is a sequence $[a_0; a_1, a_2, \dots, a_n]$. The following restrictions are applied:

- n is a non-negative finite integer,
- the elements $a_0, a_1, a_2, \dots, a_n$ are integers,
- $a_i > 0$ for each $i > 0$,
- $a_n > 1$ if $n > 0$.

These restrictions allow to establish a one-to-one correspondence between rational numbers and finite continued fractions: every rational number x corresponds to the unique continued fraction $[a_0; a_1, a_2, \dots, a_n]$ such that

$$x = a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{\ddots + \frac{1}{a_n}}}}.$$

Thus, the following notation is used: $x = [a_0; a_1, a_2, \dots, a_n]$.

Given the continued fraction for a rational number x ($0 < x < 1$), find the continued fraction for the number $(1 - x)$.

Input format

The first line of input contains an integer n ($1 \leq n \leq 100\,000$). The next line contains $(n + 1)$ integers: $a_0, a_1, a_2, \dots, a_n$.

These are the elements of the given continued fraction. It is guaranteed that the given sequence is a proper continued fraction, in other words, the given numbers satisfy all restrictions from the definition. The given continued fraction corresponds to a rational number x such that $0 < x < 1$, and for each i , $a_i \leq 100$.

Output format

On the first line of output, print an integer m , the length of the resulting continued fraction. On the next line, print $(m + 1)$ positive integers: $b_0, b_1, b_2, \dots, b_m$. These must be the elements of the resulting continued fraction. The printed sequence must be a proper continued fraction, in other words, the printed numbers must satisfy all restrictions from the definition. The following equality must hold:

$$1 - [a_0; a_1, a_2, \dots, a_n] = [b_0; b_1, b_2, \dots, b_m].$$

Sample 1

Input	Output
2 0 1 2	1 0 3
2 0 2 3	3 0 1 1 3