Codeforces Round #460 (Div. 2)

A. Supermarket

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

We often go to supermarkets to buy some fruits or vegetables, and on the tag there prints the price for a kilo. But in some supermarkets, when asked how much the items are, the clerk will say that a yuan for b kilos (You don't need to care about what "yuan" is), the same as a/b yuan for a kilo.

Now imagine you'd like to buy m kilos of apples. You've asked n supermarkets and got the prices. Find the minimum cost for those apples.

You can assume that there are enough apples in all supermarkets.

Input

The first line contains two positive integers n and m ($1 \le n \le 5\,000$, $1 \le m \le 100$), denoting that there are n supermarkets and you want to buy m kilos of apples.

The following n lines describe the information of the supermarkets. Each line contains two positive integers a, b ($1 \le a, b \le 100$), denoting that in this supermarket, you are supposed to pay a yuan for b kilos of apples.

Output

The only line, denoting the minimum cost for m kilos of apples. Please make sure that the absolute or relative error between your answer and the correct answer won't exceed 10^{-6}.

Formally, let your answer be x, and the jury's answer be y. Your answer is considered correct if $\frac{|x-y|}{\max{(1, |y|)}} \le 10^{-6}$.

Examples

input

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

output

```
1.66666667
```

input

```
2 1
99 100
98 99
```

output

```
0.98989899
```

Note

In the first sample, you are supposed to buy 5 kilos of apples in supermarket 3. The cost is $5/3$ yuan.

In the second sample, you are supposed to buy 1 kilo of apples in supermarket 2. The cost is $98/99$ yuan.

B. Perfect Number

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

We consider a positive integer perfect, if and only if the sum of its digits is exactly 10. Given a positive integer k, your task is to find the k-th smallest perfect positive integer.

Input

A single line with a positive integer k ($1 \le k \le 10\,000$).

Output

A single number, denoting the k-th smallest perfect integer.

Examples

input

```
1
```

output

```
19
```

input

```
2
```

output

```
28
```

Note

The first perfect integer is 19 and the second one is 28.

C. Seat Arrangements

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Suppose that you are in a campus and have to go for classes day by day. As you may see, when you hurry to a classroom, you surprisingly find that many seats there are already occupied. Today you and your friends went for class, and found out that some of the seats were occupied.

The classroom contains n rows of seats and there are m seats in each row. Then the classroom can be represented as an $n \times m$ matrix. The character '`.`' represents an empty seat, while '`*`' means that the seat is occupied. You need to find k consecutive empty seats in the same row or column and arrange those seats for you and your friends. Your task is to find the number of ways to arrange the seats. **Two ways are considered different if sets of places that students occupy differs.**

Input

The first line contains three positive integers n, m, k $(1 \leq n, m, k \leq 2\,000)$, where n, m represent the sizes of the classroom and k is the number of consecutive seats you need to find.

Each of the next n lines contains m characters '`.`' or '`*`'. They form a matrix representing the classroom, '`.`' denotes an empty seat, and '`*`' denotes an occupied seat.

Output

A single number, denoting the number of ways to find k empty seats in the same row or column.

Examples

input

```
2 3 2
**.
...
```

output

```
3
```

input

```
1 2 2
..
```

output

```
1
```

input

```
3 3 4
.*.
*.*
.*.
```

output

```
0
```

Note

In the first sample, there are three ways to arrange those seats. You can take the following seats for your arrangement.

- $(1, 3), (2, 3)$
- $(2, 2), (2, 3)$
- $(2, 1), (2, 2)$

D. Substring

time limit per test: 3 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

You are given a graph with n nodes and m **directed** edges. One lowercase letter is assigned to each node. We define a path's value as the number of the most frequently occurring letter. For example, if letters on a path are "`abaca`", then the value of that path is 3. Your task is find a path whose value is the largest.

Input

The first line contains two positive integers n, m $(1 \leq n, m \leq 300\,000)$, denoting that the graph has n nodes and m directed edges.

The second line contains a string s with only lowercase English letters. The i-th character is the letter assigned to the i-th node.

Then m lines follow. Each line contains two integers x, y $(1 \leq x, y \leq n)$, describing a directed edge from x to y. Note that x can be equal to y and there can be multiple edges between x and y. Also the graph can be not connected.

Output

Output a single line with a single integer denoting the largest value. If the value can be arbitrarily large, output `-1` instead.

Examples

input

```
5 4
abaca
1 2
1 3
3 4
4 5
```

output

```
3
```

input

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

output

```
-1
```

input

```
10 14
xzyzyzyzqx
1 2
2 4
3 5
4 5
2 6
6 8
6 5
2 10
3 9
10 9
4 6
1 10
2 8
3 7
```

output

```
4
```

Note

In the first sample, the path with largest value is $1 \to 3 \to 4 \to 5$. The value is 3 because the letter '`a`' appears 3 times.

E. Congruence Equation

time limit per test: 3 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

Given an integer x. Your task is to find out how many positive integers n ($1 \le n \le x$) satisfy

$$n \cdot a^n \equiv b \pmod{p},$$

where a, b, p are all known constants.

Input

The only line contains four integers a, b, p, x ($2 \le p \le 10^6+3$, $1 \le a, b < p$, $1 \le x \le 10^{12}$). It is guaranteed that p is a prime.

Output

Print a single integer: the number of possible answers n.

Examples

input

```
2 3 5 8
```

output

```
2
```

input

```
4 6 7 13
```

output

```
1
```

input

```
233 233 10007 1
```

output

```
1
```

Note

In the first sample, we can see that $n = 2$ and $n = 8$ are possible answers.

F. A Game With Numbers

time limit per test: 4 seconds

memory limit per test: 512 megabytes

input: standard input

output: standard output

Imagine that Alice is playing a card game with her friend Bob. They both have exactly 8 cards and there is an integer on each card, ranging from 0 to 4. In each round, Alice or Bob in turns choose two cards from different players, let them be a and b, where a is the number on the player's card, and b is the number on the opponent's card. It is necessary that $a \cdot b \ne 0$. Then they calculate $c = (a+b) \bmod 5$ and replace the number a with c. The player who ends up with numbers on all 8 cards being 0, wins.

Now Alice wants to know who wins in some situations. She will give you her cards' numbers, Bob's cards' numbers and the person playing the first round. Your task is to determine who wins if both of them choose the best operation in their rounds.

Input

The first line contains one positive integer T ($1 \le T \le 100\,000$), denoting the number of situations you need to consider.

The following lines describe those T situations. For each situation:

- The first line contains a non-negative integer f ($0 \le f \le 1$), where $f = 0$ means that Alice plays first and $f = 1$ means Bob plays first.
- The second line contains 8 non-negative integers $a_1, a_2, \ldots, a_8$ ($0 \le a_i \le 4$), describing Alice's cards.
- The third line contains 8 non-negative integers $b_1, b_2, \ldots, b_8$ ($0 \le b_i \le 4$), describing Bob's cards.

We guarantee that if $f = 0$, we have $\sum_{i=1}^{8} a_i \ne 0$. Also when $f = 1$, $\sum_{i=1}^{8} b_i \ne 0$ holds.

Output

Output T lines. For each situation, determine who wins. Output

- "`Alice`" (without quotes) if Alice wins.
- "`Bob`" (without quotes) if Bob wins.
- "`Deal`" (without quotes) if it gets into a deal, i.e. no one wins.

Example

input

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

```
0 0 0 1 0 0 0 0
0 0 0 0 4 0 0 0
0
1 0 0 0 0 0 0 0
0 0 0 4 0 0 2 0
1
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
```

output

```
Alice
Bob
Alice
Deal
```

Note

In the first situation, Alice has all her numbers 0. So she wins immediately.

In the second situation, Bob picks the numbers 4 and 1. Because we have $(4 + 1) \bmod 5 = 0$, Bob wins after this operation.

In the third situation, Alice picks the numbers 1 and 4. She wins after this operation.

In the fourth situation, we can prove that it falls into a loop.