A. Dreamoon and Stairs

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

Dreamoon wants to climb up a stair of n steps. He can climb 1 or 2 steps at each move. Dreamoon wants the number of moves to be a multiple of an integer m.

What is the minimal number of moves making him climb to the top of the stairs that satisfies his condition?

Input

The single line contains two space separated integers n, m ($0 < n \le 10000, 1 < m \le 10$).

Output

Print a single integer — the minimal number of moves being a multiple of m. If there is no way he can climb satisfying condition print -1 instead.

Examples

input

```
10 2
```

output

```
6
```

input

```
3 5
```

output

```
-1
```

Note

For the first sample, Dreamoon could climb in 6 moves with following sequence of steps: {2, 2, 2, 2, 1, 1}.

For the second sample, there are only three valid sequence of steps {2, 1}, {1, 2}, {1, 1, 1} with 2, 2, and 3 steps respectively. All these numbers are not multiples of 5.

B. Dreamoon and WiFi

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

Dreamoon is standing at the position 0 on a number line. Drazil is sending a list of commands through Wi-Fi to Dreamoon's smartphone and Dreamoon follows them.

Each command is one of the following two types:

1. Go 1 unit towards the positive direction, denoted as `'+'`
2. Go 1 unit towards the negative direction, denoted as `'-'`

But the Wi-Fi condition is so poor that Dreamoon's smartphone reports some of the commands can't be recognized and Dreamoon knows that some of them might even be wrong though successfully recognized. Dreamoon decides to follow every recognized command and toss a fair coin to decide those unrecognized ones (that means, he moves to the 1 unit to the negative or positive direction with the same probability 0.5).

You are given an original list of commands sent by Drazil and list received by Dreamoon. What is the probability that Dreamoon ends in the position originally supposed to be final by Drazil's commands?

Input

The first line contains a string s_1 — the commands Drazil sends to Dreamoon, this string consists of only the characters in the set `{'+', '-'}`.

The second line contains a string s_2 — the commands Dreamoon's smartphone recognizes, this string consists of only the characters in the set `{'+', '-', '?'}`. `'?'` denotes an unrecognized command.

Lengths of two strings are equal and do not exceed 10.

Output

Output a single real number corresponding to the probability. The answer will be considered correct if its relative or absolute error doesn't exceed 10^{-9}.

Examples

input

```
++-+-
+-+-+
```

output

```
1.000000000000
```

input

```
+-+-
+-??
```

output

```
0.500000000000
```

input

```
+++
??-
```

output

```
0.000000000000
```

Note

For the first sample, both s_1 and s_2 will lead Dreamoon to finish at the same position $+1$.

For the second sample, s_1 will lead Dreamoon to finish at position 0, while there are four possibilites for s_2: {"+-++", "+-+-", "+--+", "+---"} with ending position {+2, 0, 0, -2} respectively. So there are 2 correct cases out of 4, so the probability of finishing at the correct position is 0.5.

For the third sample, s_2 could only lead us to finish at positions {+1, -1, -3}, so the probability to finish at the correct position $+3$ is 0.

C. Dreamoon and Sums

time limit per test: 1.5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Dreamoon loves summing up something for no reason. One day he obtains two integers a and b occasionally. He wants to calculate the sum of all *nice* integers. Positive integer x is called *nice* if $\mathrm{mod}(x, b) \neq 0$ and $\frac{\mathrm{div}(x,b)}{\mathrm{mod}(x,b)} = k$, where k is some **integer** number in range $[1, a]$.

By $\mathrm{div}(x, y)$ we denote the *quotient* of integer division of x and y. By $\mathrm{mod}(x, y)$ we denote the *remainder* of integer division of x and y. You can read more about these operations here: `http://goo.gl/AcsXhT`.

The answer may be large, so please print its remainder modulo $1\,000\,000\,007$ ($10^9 + 7$). Can you compute it faster than Dreamoon?

Input

The single line of the input contains two integers a, b ($1 \leq a, b \leq 10^7$).

Output

Print a single integer representing the answer modulo $1\,000\,000\,007$ ($10^9 + 7$).

Examples

input

```
1 1
```

output

```
0
```

input

```
2 2
```

output

```
8
```

Note

For the first sample, there are no nice integers because $\mathrm{mod}(x, 1)$ is always zero.

For the second sample, the set of nice integers is $\{3, 5\}$.

D. Dreamoon and Sets

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

Dreamoon likes to play with sets, integers and $\gcd$. $\gcd(a, b)$ is defined as the largest positive integer that divides both a and b.

Let S be a set of exactly four distinct integers greater than 0. Define S to be of rank k if and only if for all pairs of distinct elements s_i, s_j from S, $\gcd(s_i, s_j) = k$.

Given k and n, Dreamoon wants to make up n sets of rank k using integers from 1 to m such that no integer is used in two different sets (of course you can leave some integers without use). Calculate the minimum m that makes it possible and print one possible solution.

Input

The single line of the input contains two space separated integers n, k ($1 \le n \le 10\,000, 1 \le k \le 100$).

Output

On the first line print a single integer — the minimal possible m.

On each of the next n lines print four space separated integers representing the i-th set.

Neither the order of the sets nor the order of integers within a set is important. If there are multiple possible solutions with minimal m, print any one of them.

Examples

input

```
1 1
```

output

```
5
1 2 3 5
```

input

```
2 2
```

output

```
22
2 4 6 22
14 18 10 16
```

Note

For the first example it's easy to see that set $\{1, 2, 3, 4\}$ isn't a valid set of rank 1 since $\gcd(2, 4) = 2 \neq 1$.

E. Dreamoon and Strings

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

Dreamoon has a string s and a pattern string p. He first removes exactly x characters from s obtaining string s' as a result. Then he calculates $\mathrm{occ}(s', p)$ that is defined as the maximal number of non-overlapping substrings equal to p that can be found in s'. He wants to make this number as big as possible.

More formally, let's define $\mathrm{ans}(x)$ as maximum value of $\mathrm{occ}(s', p)$ over all s' that can be obtained by removing exactly x characters from s. Dreamoon wants to know $\mathrm{ans}(x)$ for all x from 0 to $|s|$ where $|s|$ denotes the length of string s.

Input

The first line of the input contains the string s ($1 \le |s| \le 2\,000$).

The second line of the input contains the string p ($1 \le |p| \le 500$).

Both strings will only consist of lower case English letters.

Output

Print $|s| + 1$ space-separated integers in a single line representing the $\mathrm{ans}(x)$ for all x from 0 to $|s|$.

Examples

input

```
aaaaa
aa
```

output

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

input

```
axbaxxb
ab
```

output

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

Note

For the first sample, the corresponding optimal values of s' after removal 0 through $|s| = 5$ characters from s are {"**aaaa**a", "**aaaa**", "**aa**a", "**aa**", "a", ""}.

For the second sample, possible corresponding optimal values of s' are {"axbaxxb", "**ab**axxb", "axb**ab**", "**abab**", "**ab**a", "**ab**", "a", ""}.