# Problem A. Tennis

| | |
|---|---|
| Input file: | `standard input` |
| Output file: | `standard output` |
| Time limit: | 2 seconds |
| Memory limit: | 64 megabytes |

Software engineer Andrew is keen on sports, especially tennis. Andrew is not perfect player, hence sometimes he wins and sometimes he loses. Obviously, all lost matches make him very angry and hence make his primary work less efficient. Therefore, Andrew has made a conclusion "I will be less angry, if I don't take part in such matches". So he has decided to win all the times. This is very straightforward to compute the probability to win whole match knowing $p_1$ and $p_2$ – probabilities of winning a point, when you are serving and not serving, respectively. Andrew is taking part in a very important match soon. Could you please help him to compute the probability to win this game?

**Tennis rules.**

Match consists of sets, which are divided to games. The serving person is the same in each game. To win a game player should score at least 4 points, but his competitor should have at least two points less. The serving person is changed after each game. To win a set player should win 6 (6-0, 6-1, 6-2, 6-3, 6-4) or 7 games (7-5). Otherwise, if the score is 6-6, the referee announces tiebreak.

**Tiebreak rules.**

The person, who has not served last game, should serve first. After first played point serving person is changed. Then the serving person is changed after each two played points. To win tiebreak player should score at least 7 points and his competitor should score at least two points less. The winner of tiebreak wins the set.

To win a match player should win 3 sets. In case of tiebreak a serving person in next set should be a player who has not served during the first serve. Otherwise, the person who has not serve during last game should serve in the next set.

## Input

A single line contains two real numbers $p_1$ and $p_2$ — probabilities of winning a point, when you are serving and not serving, respectively. The maximum number of digits after decimal point is 4.

$0 \le p_1, p_2 \le 1$

## Output

A single line should contain the probability of winning the whole match. The absolute and relative error of given answer should not exceed $10^{-9}$.

## Examples

| standard input | standard output |
|---|---|
| 0.5 0.5 | 0.500000000000000 |
| 0.7 0.3 | 0.500000000000000 |

## Note

Andrew always serves in the first game of the first set.

# Problem B. Traveling Salesman Problem

| | |
|---|---|
| Input file: | `standard input` |
| Output file: | `standard output` |
| Time limit: | 2 seconds |
| Memory limit: | 64 megabytes |

Nowadays British scientists has been investigating the opportunities to increase the performance of traditional computers. Recent research has shown that changing logical zero to logical right shift ($x = x$ / 2) and logical one to logical negation ($x = 1$ - $x$) will lead to a significant architecture enhancement. Thus, new architecture will be able to solve some NP hard problems. Assume that mentioned above basic operations will be new zero and new one, respectively.

For example, well known traveling salesman problem solved in the proposed architecture will be equivalent to obtaining number (fraction) $\frac{a}{2^b}$ from a number of one (initial state). Therefore, you are required to write a program (sequence of zeros and ones) to solve traveling salesman problem for given $a$ and $b$.

## Input

Single line contains two integer numbers $a$ and $b$.

$1 \le b \le 60$,

$1 \le a < 2^b$,

$a$ is odd.

## Output

Single line should contain a binary sequence, which is a solution for traveling salesman problem using new architecture. If there are multiple solutions, the minimal length program should be printed.

## Examples

| standard input | standard output |
|---|---|
| 1 1 | 0 |
| 3 3 | 0010 |

# Problem C. Maya's message

| | |
|---|---|
| Input file: | `standard input` |
| Output file: | `standard output` |
| Time limit: | 2 seconds |
| Memory limit: | 64 megabytes |

This is a well known fact that Maya's message regarding Apocalypse on 21 December 2012 has not happened. Perhaps, the issue was in a wrong way of the message decryption. Despite Maya's calendar is quite complicated, researchers have managed to compute the exact number of days before Apocalypse event.

Unfortunately, this number has two types of digits, i.e. some of the are easy readable, but some of them are quite blurred. Maya society used to have a cult of number of $k$ during those days. Therefore, scientists are sure that number from their message is a multiple of $k$. So the government is quite worried about this message. Hence, the task is to find the minimal possible number, which could be mentioned by Maya.

## Input

First line contains two integer numbers $n$ and $k$ — length of the message and cult number, respectively.

Second line contains the message, which has total $n$ digits and "?" signs stand for clear and blurry symbols, respectively.

$1 \leq n, k \leq 1000$.

## Output

Single line should contain minimal possible number, which has exactly $n$ digits – the message of Maya. Note, that ancient Maya don't like leading zeros in numbers. Therefore, there is no chance that the message contains any of them.

If it is impossible to recover the message under given constraints, the line should contain -1.

## Examples

| standard input | standard output |
|---|---|
| 2 14<br>4? | 42 |
| 2 17<br>4? | -1 |

# Problem D. Viktoria and resentments

| | |
|---|---|
| Input file: | `standard input` |
| Output file: | `standard output` |
| Time limit: | 2 seconds |
| Memory limit: | 256 megabytes |

Viktoria is so easily offended. She was also offended by Alex during their walk. It is known that Alex and Viktoria were walking $n$ minutes. And every minute Viktoria was either offended or not offended by Alex. Since we are all programmers, let's denote these $n$ minutes of Viktoria's state as a string of $n$ bits. So, the $i$-th bit of the string is `1` if she was offended in the $i$-th minute, otherwise this bit is `0`.

Alex considers that Viktoria takes offense at him too often. And he likes to prove everything. That's why Alex wants to count the number of time periods where her offence was **strictly greater** than $X$. A pair of numbers $(l, r)$, where $1 \le l \le r \le n$, is considered to be a time period. Viktoria's offense during some period is a ratio of the amount of minutes she took offense to the total amount of minutes in a period. More formally, Viktoria's offense during the period $(l, r)$ is $\frac{cnt_1}{cnt_{all}}$, where $cnt_1$ is an amount of ones in the segment $(l, r)$ of the bit string, and $cnt_{all}$ is the total amount of characters in the bit string from the $l$-th position to the $r$-th position.

Help Alex to prove to Viktoria that she takes offense too often!

## Input

The first line contains the description of $X$ in the form of two space-separated integers: $P$ and $Q$. ($0 \le P \le 1000$, $1 \le Q \le 1000$, $P \le Q$), i. e. $X$ is equal to $\frac{P}{Q}$.

The next line contains a string $s$. It consists of the characters '`0`' and '`1`'. The length of the string is positive and doesn't exceed 200 000.

## Output

Output the number of segments $(l, r)$, where Viktoria's offense is strictly greater than $X$.

## Examples

| standard input | standard output |
|---|---|
| 3 10<br>10110 | 13 |
| 9 10<br>010100101101 | 7 |

# Problem E. Kth subtree

| | |
|---|---|
| Input file: | `standard input` |
| Output file: | `standard output` |
| Time limit: | 2 seconds |
| Memory limit: | 256 megabytes |

Legends for tasks are different. There are long or short. There are boring or funny. There are understandable or not. You decide what this. Given a tree from $n$ vertices. Also given are $q$ queries. One query is a pair of numbers $v$ and $k$. We choose $v$ as a root and write down the sizes of the subtrees of each vertex of the tree in the nondecreasing order. The answer to the query is the $k$ th number in this list.

## Input

The first line of input contains two numbers - $n$ and $q$ ($1 \le n \le 10^5$, $1 \le q \le 10^5$). Each of the following $n - 1$ rows contains a description of the edge of the tree - its ends $u_i$ and $v_i$ ($1 \le u_i, v_i \le n$) Each of the following $q$ lines contains a description of the next query - the numbers $v_i$ and $k_i$ ($1 \le v_i, k_i \le n$)

## Output

Output a response to each of the requests on a separate line.

## Example

| standard input | standard output |
|---|---|
| 5 5<br>1 2<br>1 3<br>1 4<br>4 5<br>2 5<br>1 3<br>4 4<br>2 3<br>2 2 | 5<br>1<br>3<br>2<br>1 |

# Problem F. A+B

| | |
|---|---|
| Input file: | `standard input` |
| Output file: | `standard output` |
| Time limit: | 2 seconds |
| Memory limit: | 64 megabytes |

Like every Andrew, our Andrew likes playing with numbers and addition in columns. He has two decimal numbers $X$ and $Y$. Initially, they are equal to 0. Number positions are numbered sequentially from left to right, numbering is 1-based. Andrew has the following types of operations:

- 1 $k$ $a$ $b$ - To insert the digit $a$ ($0 \le a \le 9$) after the $k$-th digit of the number $X$ (if $k$ is 0, then the digit goes to the beginning of the number). Similarly insert the digit $b$ in the number $Y$. It is guaranteed that at the moment of the execution of operation the numbers contain at least $k$ digits.
- 2 $k$ - To delete the $k$-th digit from the numbers $X$ and $Y$. It is guaranteed that at the moment of the execution of operation the numbers contain at least $k$ digits.
- 3 $k$ - To output the $k$-th digit of the number $Z$. $Z$ is equal to the sum of $X$ and $Y$. Note that it is not guaranteed for this operation that the number $Z$ has the $k$ -th digit. If there is no this digit, output $-1$.

It is guaranteed that after performing each operation the numbers $X$ and $Y$ do not have leading zeros.

## Input

The first line of input contains a single number $n$ ($1 \le n \le 3 \times 10^5$). It is an amount of performed operations. The following $n$ lines contain descriptions of operations. The format is described above. For all operations the value of parameter $k$ satisfies the inequality $0 \le k \le n$.

## Output

For each third type operation output its result on a separate line.

## Example

| standard input | standard output |
|---|---|
| 12<br>1 0 8 1<br>3 1<br>3 2<br>2 1<br>1 0 8 3<br>3 1<br>3 2<br>1 0 4 5<br>3 1<br>3 2<br>3 3<br>3 5 | 9<br>0<br>1<br>1<br>1<br>0<br>1<br>-1 |

# Problem G. Many dimensional dice

| | |
|---|---|
| Input file: | `standard input` |
| Output file: | `standard output` |
| Time limit: | 2 seconds |
| Memory limit: | 64 megabytes |

All Andrews in general like to play different games. This Andrew is not an exception and he likes to do it as well. Recently, he has attended a casino under some unknown circumstances. Croupier has offered him to play some very attractive game.

You are given a dice with $k$ faces. Each of them has a unique number from 1 to $k$. You are allowed to roll a dice no more $n$ times and stop after some arbitrary chosen round. The amount of money you win is equals to the last number appeared on the dice top face. All of the outcomes after rolling a dice have equal probabilities of appearance.

Andrew is a quite strong mathematician and hence knows the optimal strategy to maximize the award. Your task is to compute the expected value of the award if you are playing like Andrew, i.e. following the optimal strategy.

## Input

A single line contains two integers $n$ and $k$.

$1 \le n, 1 \le k, n \times k \le 10^{12}$

## Output

A single line should contain a single real number – expected amount of money you may win following the optimal strategy. Absolute and relative error should not exceed $10^{-9}$.

## Examples

| standard input | standard output |
|---|---|
| 1 6 | 3.500000000000000 |
| 2 6 | 4.250000000000000 |

# Problem H. 3XOR

| | |
|---|---|
| Input file: | `standard input` |
| Output file: | `standard output` |
| Time limit: | 1 second |
| Memory limit: | 64 megabytes |

Like every Andrew, our Andrew likes arrays. And he likes to play with arrays and search for different interesting numbers. Once he got an array of integers $A$. The length of the array was $n$. Andrew decided to choose three indexes (possibly identical to each other) $i$, $j$, $k$ and then calculated the bitwise exclusive or of numbers $A[i]$, $A[j]$ and $A[k]$. Andrew even invented the name 3XOR for this operation. After the euphoria passed, he wondered what would happen if you took all possible unordered combinations of the indexes $i$, $j$, $k$, used a new operation and then added all the results. It must be some number. Help Andrew solve this problem!

## Input

The first line of input contains a number $n$. The next line contains array $A$.

$3 \le n \le 10^4$

$0 \le A[i] \le 10^6$

## Output

Just output the answer

## Examples

| standard input | standard output |
|---|---|
| 3<br>1 2 3 | 42 |
| 5<br>42 4 42 2 42 | 2868 |