A. Memory and Crow

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

There are n integers $b_1, b_2, ..., b_n$ written in a row. For all i from 1 to n, values a_i are defined by the crows performing the following procedure:

- The crow sets a_i initially 0.
- The crow then adds b_i to a_i, subtracts b_{i+1}, adds the b_{i+2} number, and so on until the n'th number. Thus, $a_i = b_i - b_{i+1} + b_{i+2} - b_{i+3}...$.

Memory gives you the values $a_1, a_2, ..., a_n$, and he now wants you to find the initial numbers $b_1, b_2, ..., b_n$ written in the row? Can you do it?

Input

The first line of the input contains a single integer n $(2 \le n \le 100\,000)$ — the number of integers written in the row.

The next line contains n, the i'th of which is a_i $(-10^9 \le a_i \le 10^9)$ — the value of the i'th number.

Output

Print n integers corresponding to the sequence $b_1, b_2, ..., b_n$. It's guaranteed that the answer is unique and fits in 32-bit integer type.

Examples

input

```
5
6 -4 8 -2 3
```

output

```
2 4 6 1 3
```

input

```
5
3 -2 -1 5 6
```

output

```
1 -3 4 11 6
```

Note

In the first sample test, the crows report the numbers $6, -4, 8, -2$, and 3 when he starts at indices $1, 2, 3, 4$ and 5 respectively. It is easy to check that the sequence $2\ 4\ 6\ 1\ 3$ satisfies the reports. For example, $6 = 2 - 4 + 6 - 1 + 3$, and $-4 = 4 - 6 + 1 - 3$.

In the second sample test, the sequence $1, -3, 4, 11, 6$ satisfies the reports. For example, $5 = 11 - 6$ and $6 = 6$.

B. Memory and Trident

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

Memory is performing a walk on the two-dimensional plane, starting at the origin. He is given a string s with his directions for motion:

- An 'L' indicates he should move one unit left.
- An 'R' indicates he should move one unit right.
- A 'U' indicates he should move one unit up.
- A 'D' indicates he should move one unit down.

But now Memory wants to end at the origin. To do this, he has a special trident. This trident can replace any character in s with any of 'L', 'R', 'U', or 'D'. However, because he doesn't want to wear out the trident, he wants to make the minimum number of edits possible. Please tell Memory what is the minimum number of changes he needs to make to produce a string that, when walked, will end at the origin, or if there is no such string.

Input

The first and only line contains the string s $(1 \le |s| \le 100\,000)$ — the instructions Memory is given.

Output

If there is a string satisfying the conditions, output a single integer — the minimum number of edits required. In case it's not possible to change the sequence in such a way that it will bring Memory to to the origin, output -1.

Examples

input
```
RRU
```
output
```
-1
```

input
```
UDUR
```
output
```
1
```

input
```
RUUR
```
output
```
2
```

Note

In the first sample test, Memory is told to walk right, then right, then up. It is easy to see that it is impossible to edit these instructions to form a valid walk.

In the second sample test, Memory is told to walk up, then down, then up, then right. One possible solution is to change s to "LDUR". This string uses 1 edit, which is the minimum possible. It also ends at the origin.

C. Memory and De-Evolution

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

Memory is now interested in the de-evolution of objects, specifically triangles. He starts with an equilateral triangle of side length x, and he wishes to perform operations to obtain an equilateral triangle of side length y.

In a single second, he can modify the length of a single side of the current triangle such that it remains a non-degenerate triangle (triangle of positive area). At any moment of time, the length of each side should be integer.

What is the minimum number of seconds required for Memory to obtain the equilateral triangle of side length y?

Input

The first and only line contains two integers x and y ($3 \le y < x \le 100\,000$) — the starting and ending equilateral triangle side lengths respectively.

Output

Print a single integer — the minimum number of seconds required for Memory to obtain the equilateral triangle of side length y if he starts with the equilateral triangle of side length x.

Examples

input
```
6 3
```
output
```
4
```

input
```
8 5
```
output
```
3
```

input
```
22 4
```
output
```
6
```

Note

In the first sample test, Memory starts with an equilateral triangle of side length 6 and wants one of side length 3. Denote a triangle with sides a, b, and c as (a, b, c). Then, Memory can do .

In the second sample test, Memory can do .

In the third sample test, Memory can do:

.

D. Memory and Scores

time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Memory and his friend Lexa are competing to get higher score in one popular computer game. Memory starts with score a and Lexa starts with score b. In a single turn, both Memory and Lexa get some integer in the range $[-k;k]$ (i.e. one integer among $-k, -k+1, -k+2, ..., -2, -1, 0, 1, 2, ..., k-1, k$) and add them to their current scores. The game has exactly t turns. Memory and Lexa, however, are not good at this game, so they both always get a random integer at their turn.

Memory wonders how many possible games exist such that he ends with a strictly higher score than Lexa. Two games are considered to be different if in at least one turn at least one player gets different score. There are $(2k+1)^{2t}$ games in total. Since the answer can be very large, you should print it modulo 10^9+7. Please solve this problem for Memory.

Input

The first and only line of input contains the four integers a, b, k, and t ($1 \le a, b \le 100$, $1 \le k \le 1000$, $1 \le t \le 100$) — the amount Memory and Lexa start with, the number k, and the number of turns respectively.

Output

Print the number of possible games satisfying the conditions modulo $1\,000\,000\,007$ (10^9+7) in one line.

Examples

input
```
1 2 2 1
```
output
```
6
```

input
```
1 1 1 2
```
output
```
31
```

input
```
2 12 3 1
```
output
```
0
```

Note

In the first sample test, Memory starts with 1 and Lexa starts with 2. If Lexa picks -2, Memory can pick 0, 1, or 2 to win. If Lexa picks -1, Memory can pick 1 or 2 to win. If Lexa picks 0, Memory can pick 2 to win. If Lexa picks 1 or 2, Memory cannot win. Thus, there are $3+2+1=6$ possible games in which Memory wins.

E. Memory and Casinos

time limit per test: 4 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

There are n casinos lined in a row. If Memory plays at casino i, he has probability p_i to win and move to the casino on the right ($i + 1$) or exit the row (if $i = n$), and a probability $1 - p_i$ to lose and move to the casino on the left ($i - 1$) or also exit the row (if $i = 1$).

We say that Memory *dominates* on the interval $i \dots j$ if he completes a walk such that,

- He starts on casino i.
- He never looses in casino i.
- He finishes his walk by winning in casino j.

Note that Memory can still walk left of the 1-st casino and right of the casino n and that always finishes the process.

Now Memory has some requests, in one of the following forms:

- $1\ i\ a\ b$: Set .
- $2\ l\ r$: Print the probability that Memory will *dominate* on the interval $l \dots r$, i.e. compute the probability that Memory will first leave the segment $l \dots r$ after winning at casino r, if she starts in casino l.

It is guaranteed that at any moment of time p is a **non-decreasing sequence**, i.e. $p_i \le p_{i+1}$ for all i from 1 to $n - 1$.

Please help Memory by answering all his requests!

Input

The first line of the input contains two integers n and $q (1 \le n, q \le 100\ 000)$, — number of casinos and number of requests respectively.

The next n lines each contain integers a_i and b_i $(1 \le a_i < b_i \le 10^9)$ — is the probability p_i of winning in casino i.

The next q lines each contain queries of one of the types specified above ($1 \le a < b \le 10^9$, $1 \le i \le n$, $1 \le l \le r \le n$).

It's guaranteed that there will be at least one query of type 2, i.e. the output will be non-empty. Additionally, it is guaranteed that p forms a non-decreasing sequence at all times.

Output

Print a real number for every request of type 2 — the probability that boy will "dominate" on that interval. Your answer will be considered correct if its absolute error does not exceed 10^{-4}.

Namely: let's assume that one of your answers is a, and the corresponding answer of the jury is b. The checker program will consider your answer correct if $|a - b| \le 10^{-4}$.

Example

input

```
3 13
1 3
1 2
2 3
2 1 1
2 1 2
2 1 3
2 2 2
2 2 3
2 3 3
1 2 2 3
2 1 1
2 1 2
2 1 3
2 2 2
2 2 3
2 3 3
```

output

```
0.3333333333
0.2000000000
0.1666666667
0.5000000000
0.4000000000
0.6666666667
0.3333333333
0.2500000000
0.2222222222
0.6666666667
0.5714285714
0.6666666667
```