Codeforces Round #350 (Div. 2)

A. Holidays

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

On the planet Mars a year lasts exactly n days (there are no leap years on Mars). But Martians have the same weeks as earthlings — 5 work days and then 2 days off. Your task is to determine the minimum possible and the maximum possible number of days off per year on Mars.

Input
The first line of the input contains a positive integer n ($1 \le n \le 1\,000\,000$) — the number of days in a year on Mars.

Output
Print two integers — the minimum possible and the maximum possible number of days off per year on Mars.

Examples

input
14
output
4 4

input
2
output
0 2

Note
In the first sample there are 14 days in a year on Mars, and therefore independently of the day a year starts with there will be exactly 4 days off .

In the second sample there are only 2 days in a year on Mars, and they can both be either work days or days off.

B. Game of Robots

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

In late autumn evening n robots gathered in the cheerful company of friends. Each robot has a unique identifier — an integer from 1 to 10^9.

At some moment, robots decided to play the game "Snowball". Below there are the rules of this game. First, all robots stand in a row. Then the first robot says his identifier. After that the second robot says the identifier of the first robot and then says his own identifier. Then the third robot says the identifier of the first robot, then says the identifier of the second robot and after that says his own. This process continues from left to right until the n-th robot says his identifier.

Your task is to determine the k-th identifier to be pronounced.

Input

The first line contains two positive integers n and k ($1 \le n \le 100\,000$, $1 \le k \le min(2 \cdot 10^9, n \cdot (n+1) / 2)$).

The second line contains the sequence $id_1, id_2, ..., id_n$ ($1 \le id_i \le 10^9$) — identifiers of roborts. It is guaranteed that all identifiers are different.

Output

Print the k-th pronounced identifier (assume that the numeration starts from 1).

Examples

input
2 2 1 2

output
1

input
4 5 10 4 18 3

output
4

Note

In the first sample identifiers of robots will be pronounced in the following order: $1, 1, 2$. As $k = 2$, the answer equals to 1.

In the second test case identifiers of robots will be pronounced in the following order: $10, 10, 4, 10, 4, 18, 10, 4, 18, 3$. As $k = 5$, the answer equals to 4.

C. Cinema

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

Moscow is hosting a major international conference, which is attended by n scientists from different countries. Each of the scientists knows exactly one language. For convenience, we enumerate all languages of the world with integers from 1 to 10^9.

In the evening after the conference, all n scientists decided to go to the cinema. There are m movies in the cinema they came to. Each of the movies is characterized by two **distinct** numbers — the index of audio language and the index of subtitles language. The scientist, who came to the movie, will be *very pleased* if he knows the audio language of the movie, will be *almost satisfied* if he knows the language of subtitles and will be *not satisfied* if he does not know neither one nor the other (note that the audio language and the subtitles language for each movie are always different).

Scientists decided to go together to the same movie. You have to help them choose the movie, such that the number of very pleased scientists is maximum possible. If there are several such movies, select among them one that will maximize the number of almost satisfied scientists.

Input

The first line of the input contains a positive integer n ($1 \le n \le 200\,000$) — the number of scientists.

The second line contains n positive integers $a_1, a_2, ..., a_n$ ($1 \le a_i \le 10^9$), where a_i is the index of a language, which the i-th scientist knows.

The third line contains a positive integer m ($1 \le m \le 200\,000$) — the number of movies in the cinema.

The fourth line contains m positive integers $b_1, b_2, ..., b_m$ ($1 \le b_j \le 10^9$), where b_j is the index of the audio language of the j-th movie.

The fifth line contains m positive integers $c_1, c_2, ..., c_m$ ($1 \le c_j \le 10^9$), where c_j is the index of subtitles language of the j-th movie.

It is guaranteed that audio languages and subtitles language are different for each movie, that is $b_j \ne c_j$.

Output

Print the single integer — the index of a movie to which scientists should go. After viewing this movie the number of very pleased scientists should be maximum possible. If in the cinema there are several such movies, you need to choose among them one, after viewing which there will be the maximum possible number of almost satisfied scientists.

If there are several possible answers print any of them.

Examples

input
3 2 3 2 2 3 2 2 3
output
2

input
6 6 3 1 1 3 7 5 1 2 3 4 5 2 3 4 5 1
output
1

Note

In the first sample, scientists must go to the movie with the index 2, as in such case the 1-th and the 3-rd scientists will be very pleased and the 2-nd scientist will be almost satisfied.

In the second test case scientists can go either to the movie with the index 1 or the index 3. After viewing any of these movies exactly **two** scientists will be very pleased and all the others will be not satisfied.

D1. Magic Powder - 1

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

This problem is given in two versions that differ only by constraints. If you can solve this problem in large constraints, then you can just write a single solution to the both versions. If you find the problem too difficult in large constraints, you can write solution to the simplified version only.

Waking up in the morning, Apollinaria decided to bake cookies. To bake one cookie, she needs n ingredients, and for each ingredient she knows the value a_i — how many grams of this ingredient one needs to bake a cookie. To prepare one cookie Apollinaria needs to use all n ingredients.

Apollinaria has b_i gram of the i-th ingredient. Also she has k grams of a magic powder. Each gram of magic powder can be turned to exactly 1 gram of any of the n ingredients and can be used for baking cookies.

Your task is to determine the maximum number of cookies, which Apollinaria is able to bake using the ingredients that she has and the magic powder.

Input

The first line of the input contains two positive integers n and k ($1 \le n, k \le 1000$) — the number of ingredients and the number of grams of the magic powder.

The second line contains the sequence $a_1, a_2, ..., a_n$ ($1 \le a_i \le 1000$), where the i-th number is equal to the number of grams of the i-th ingredient, needed to bake one cookie.

The third line contains the sequence $b_1, b_2, ..., b_n$ ($1 \le b_i \le 1000$), where the i-th number is equal to the number of grams of the i-th ingredient, which Apollinaria has.

Output

Print the maximum number of cookies, which Apollinaria will be able to bake using the ingredients that she has and the magic powder.

Examples

input
3 1 2 1 4 11 3 16
output
4

input
4 3 4 3 5 6 11 12 14 20
output
3

Note

In the first sample it is profitably for Apollinaria to make the existing 1 gram of her magic powder to ingredient with the index 2, then Apollinaria will be able to bake 4 cookies.

In the second sample Apollinaria should turn 1 gram of magic powder to ingredient with the index 1 and 1 gram of magic powder to ingredient with the index 3. Then Apollinaria will be able to bake 3 cookies. The remaining 1 gram of the magic powder can be left, because it can't be used to increase the answer.

D2. Magic Powder - 2

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

The term of this problem is the same as the previous one, the only exception — increased restrictions.

Input

The first line contains two positive integers n and k ($1 \le n \le 100\,000$, $1 \le k \le 10^9$) — the number of ingredients and the number of grams of the magic powder.

The second line contains the sequence $a_1, a_2, ..., a_n$ ($1 \le a_i \le 10^9$), where the i-th number is equal to the number of grams of the i-th ingredient, needed to bake one cookie.

The third line contains the sequence $b_1, b_2, ..., b_n$ ($1 \le b_i \le 10^9$), where the i-th number is equal to the number of grams of the i-th ingredient, which Apollinaria has.

Output

Print the maximum number of cookies, which Apollinaria will be able to bake using the ingredients that she has and the magic powder.

Examples

input
1 1000000000 1 1000000000
output
2000000000

input
10 1 1000000000 1000000000 1000000000 1000000000 1000000000 1000000000 1000000000 1000000000 1000000000 1000000000 1 1 1 1 1 1 1 1 1 1
output
0

input
3 1 2 1 4 11 3 16
output
4

input
4 3 4 3 5 6 11 12 14 20
output
3

E. Correct Bracket Sequence Editor

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

Recently Polycarp started to develop a text editor that works only with correct bracket sequences (abbreviated as CBS).

Note that a bracket sequence is correct if it is possible to get a correct mathematical expression by adding "+"-s and "1"-s to it. For example, sequences "(()())()", "()" and "(()(()))" are correct, while ")(", "(()" and "(()))(" are not. Each bracket in CBS has a pair. For example, in "(()(()))":

- 1st bracket is paired with 8th,
- 2d bracket is paired with 3d,
- 3d bracket is paired with 2d,
- 4th bracket is paired with 7th,
- 5th bracket is paired with 6th,
- 6th bracket is paired with 5th,
- 7th bracket is paired with 4th,
- 8th bracket is paired with 1st.

Polycarp's editor currently supports only three operations during the use of CBS. The cursor in the editor takes the whole position of one of the brackets (not the position between the brackets!). There are three operations being supported:

- «L» — move the cursor one position to the left,
- «R» — move the cursor one position to the right,
- «D» — delete the bracket in which the cursor is located, delete the bracket it's paired to and all brackets between them (that is, delete a substring between the bracket in which the cursor is located and the one it's paired to).

After the operation "D" the cursor moves to the nearest bracket to the right (of course, among the non-deleted). If there is no such bracket (that is, the suffix of the CBS was deleted), then the cursor moves to the nearest bracket to the left (of course, among the non-deleted).

There are pictures illustrated several usages of operation "D" below.

All incorrect operations (shift cursor over the end of CBS, delete the whole CBS, etc.) are not supported by Polycarp's editor.

Polycarp is very proud of his development, can you implement the functionality of his editor?

Input

The first line contains three positive integers n, m and p ($2 \le n \le 500\,000$, $1 \le m \le 500\,000$, $1 \le p \le n$) — the number of brackets in the correct bracket sequence, the number of operations and the initial position of cursor. Positions in the sequence are numbered from left to right, starting from one. It is guaranteed that n is even.

It is followed by the string of n characters "(" and ")" forming the correct bracket sequence.

Then follow a string of m characters "L", "R" and "D" — a sequence of the operations. Operations are carried out one by one from the first to the last. It is guaranteed that the given operations never move the cursor outside the bracket sequence, as well as the fact that after all operations a bracket sequence will be non-empty.

Output

Print the correct bracket sequence, obtained as a result of applying all operations to the initial sequence.

Examples

input
8 4 5 (())()() RDLD
output
()

input
12 5 3 ((()())(())) RRDLD
output
(()(()))

input
8 8 8 (())()() LLLLLLDD

output
()()

Note

In the first sample the cursor is initially at position 5. Consider actions of the editor:

1. command "R" — the cursor moves to the position 6 on the right;
2. command "D" — the deletion of brackets from the position 5 to the position 6. After that CBS takes the form (())(), the cursor is at the position 5;
3. command "L" — the cursor moves to the position 4 on the left;
4. command "D" — the deletion of brackets from the position 1 to the position 4. After that CBS takes the form (), the cursor is at the position 1.

Thus, the answer is equal to ().

F. Restore a Number

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

Vasya decided to pass a very large integer n to Kate. First, he wrote that number as a string, then he appended to the right integer k — the number of digits in n.

Magically, all the numbers were shuffled in arbitrary order while this note was passed to Kate. The only thing that Vasya remembers, is a non-empty substring of n (a substring of n is a sequence of consecutive digits of the number n).

Vasya knows that there may be more than one way to restore the number n. Your task is to find the smallest possible initial integer n. Note that decimal representation of number n contained no leading zeroes, except the case the integer n was equal to zero itself (in this case a single digit `0` was used).

Input

The first line of the input contains the string received by Kate. The number of digits in this string does not exceed $1\,000\,000$.

The second line contains the substring of n which Vasya remembers. This string can contain leading zeroes.

It is guaranteed that the input data is correct, and the answer always exists.

Output

Print the smalles integer n which Vasya could pass to Kate.

Examples

input
003512 021
output
30021

input
199966633300 63
output
3036366999

Codeforces (c) Copyright 2010-2016 Mike Mirzayanov
The only programming contests Web 2.0 platform