Codeforces Round #301 (Div. 2)

A. Combination Lock

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

Scrooge McDuck keeps his most treasured savings in a home safe with a combination lock. Each time he wants to put there the treasures that he's earned fair and square, he has to open the lock.

The combination lock is represented by n rotating disks with digits from 0 to 9 written on them. Scrooge McDuck has to turn some disks so that the combination of digits on the disks forms a secret combination. In one move, he can rotate one disk one digit forwards or backwards. In particular, in one move he can go from digit 0 to digit 9 and vice versa. What minimum number of actions does he need for that?

Input

The first line contains a single integer n ($1 \le n \le 1000$) — the number of disks on the combination lock.

The second line contains a string of n digits — the original state of the disks.

The third line contains a string of n digits — Scrooge McDuck's combination that opens the lock.

Output

Print a single integer — the minimum number of moves Scrooge McDuck needs to open the lock.

Sample test(s)

input
5 82195 64723

output
13

Note

In the sample he needs 13 moves:

- 1 disk: $8 \to 7 \to 6$
- 2 disk: $2 \to 3 \to 4$
- 3 disk: $1 \to 0 \to 9 \to 8 \to 7$
- 4 disk: $9 \to 0 \to 1 \to 2$
- 5 disk: $5 \to 4 \to 3$

B. School Marks

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

Little Vova studies programming in an elite school. Vova and his classmates are supposed to write n progress tests, for each test they will get a mark from 1 to p. Vova is very smart and he can write every test for any mark, but he doesn't want to stand out from the crowd too much. If the sum of his marks for all tests exceeds value x, then his classmates notice how smart he is and start distracting him asking to let them copy his homework. And if the median of his marks will be lower than y points *(the definition of a median is given in the notes)*, then his mom will decide that he gets too many bad marks and forbid him to play computer games.

Vova has already wrote k tests and got marks $a_1, ..., a_k$. He doesn't want to get into the first or the second situation described above and now he needs to determine which marks he needs to get for the remaining tests. Help him do that.

Input

The first line contains 5 space-separated integers: n, k, p, x and y ($1 \le n \le 999$, n is odd, $0 \le k < n$, $1 \le p \le 1000$, $n \le x \le n \cdot p$, $1 \le y \le p$). Here n is the number of tests that Vova is planned to write, k is the number of tests he has already written, p is the maximum possible mark for a test, x is the maximum total number of points so that the classmates don't yet disturb Vova, y is the minimum median point so that mom still lets him play computer games.

The second line contains k space-separated integers: $a_1, ..., a_k$ ($1 \le a_i \le p$) — the marks that Vova got for the tests he has already written.

Output

If Vova cannot achieve the desired result, print "`-1`".

Otherwise, print $n - k$ space-separated integers — the marks that Vova should get for the remaining tests. If there are multiple possible solutions, print any of them.

Sample test(s)

input
5 3 5 18 4 3 5 4
output
4 1

input
5 3 5 16 4 5 5 5
output
-1

Note

The median of sequence $a_1, ..., a_n$ where n is odd (in this problem n is always odd) is the element staying on $(n + 1) / 2$ position in the sorted list of a_i.

In the first sample the sum of marks equals 3 + 5 + 4 + 4 + 1 = 17, what doesn't exceed 18, that means that Vova won't be disturbed by his classmates. And the median point of the sequence {1, 3, 4, 4, 5} equals to 4, that isn't less than 4, so his mom lets him play computer games.

Please note that you do not have to maximize the sum of marks or the median mark. Any of the answers: "`4 2`", "`2 4`", "`5 1`", "`1 5`", "`4 1`", "`1 4`" for the first test is correct.

In the second sample Vova got three '5' marks, so even if he gets two '1' marks, the sum of marks will be 17, that is more than the required value of 16. So, the answer to this test is "`-1`".

C. Ice Cave

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

You play a computer game. Your character stands on some level of a multilevel ice cave. In order to move on forward, you need to descend one level lower and the only way to do this is to fall through the ice.

The level of the cave where you are is a rectangular square grid of n rows and m columns. Each cell consists either from intact or from cracked ice. From each cell you can move to cells that are side-adjacent with yours (due to some limitations of the game engine you cannot make jumps on the same place, i.e. jump from a cell to itself). If you move to the cell with cracked ice, then your character falls down through it and if you move to the cell with intact ice, then the ice on this cell becomes cracked.

Let's number the rows with integers from 1 to n from top to bottom and the columns with integers from 1 to m from left to right. Let's denote a cell on the intersection of the r-th row and the c-th column as (r, c).

You are staying in the cell (r_1, c_1) and this cell is cracked because you've just fallen here from a higher level. You need to fall down through the cell (r_2, c_2) since the exit to the next level is there. Can you do this?

Input

The first line contains two integers, n and m $(1 \le n, m \le 500)$ — the number of rows and columns in the cave description.

Each of the next n lines describes the initial state of the level of the cave, each line consists of m characters "`.`" (that is, intact ice) and "`X`" (cracked ice).

The next line contains two integers, r_1 and c_1 $(1 \le r_1 \le n, 1 \le c_1 \le m)$ — your initial coordinates. It is guaranteed that the description of the cave contains character '`X`' in cell (r_1, c_1), that is, the ice on the starting cell is initially cracked.

The next line contains two integers r_2 and c_2 $(1 \le r_2 \le n, 1 \le c_2 \le m)$ — the coordinates of the cell through which you need to fall. The final cell may coincide with the starting one.

Output

If you can reach the destination, print '`YES`', otherwise print '`NO`'.

Sample test(s)

input

```
4 6
X...XX
...XX.
.X..X.
......
1 6
2 2
```

output

```
YES
```

input

```
5 4
.X..
...X
X.X.
....
.XX.
5 3
1 1
```

output

```
NO
```

input

```
4 7
..X.XX.
.XX..X.
X...X..
X......
2 2
1 6
```

output

```
YES
```

Note

In the first sample test one possible path is:

$$(1,6) \to (2,6) \to (3,6) \to (4,6) \to (4,5) \to (4,4) \to (4,3) \to (4,2) \to (4,1) \to (3,1) \to (2,1) \to (2,2) \to (2,3) \to (1,3) \to (1,2) \to (2,2)$$

After the first visit of cell $(2, 2)$ the ice on it cracks and when you step there for the second time, your character falls through the ice as intended.

D. Bad Luck Island

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

The Bad Luck Island is inhabited by three kinds of species: r rocks, s scissors and p papers. At some moments of time two random individuals meet (all pairs of individuals can meet equiprobably), and if they belong to different species, then one individual kills the other one: a rock kills scissors, scissors kill paper, and paper kills a rock. Your task is to determine for each species what is the probability that this species will be the only one to inhabit this island after a long enough period of time.

Input

The single line contains three integers r, s and p ($1 \le r, s, p \le 100$) — the original number of individuals in the species of rock, scissors and paper, respectively.

Output

Print three space-separated real numbers: the probabilities, at which the rocks, the scissors and the paper will be the only surviving species, respectively. The answer will be considered correct if the relative or absolute error of each number doesn't exceed 10^{-9}.

Sample test(s)

input
2 2 2
output
0.333333333333 0.333333333333 0.333333333333

input
2 1 2
output
0.150000000000 0.300000000000 0.550000000000

input
1 1 3
output
0.057142857143 0.657142857143 0.285714285714

E. Infinite Inversions

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

There is an infinite sequence consisting of all positive integers in the increasing order: $p = \{1, 2, 3, ...\}$. We performed n *swap* operations with this sequence. A $swap(a, b)$ is an operation of swapping the elements of the sequence on positions a and b. Your task is to find the number of inversions in the resulting sequence, i.e. the number of such index pairs (i, j), that $i < j$ and $p_i > p_j$.

Input

The first line contains a single integer n ($1 \le n \le 10^5$) — the number of *swap* operations applied to the sequence.

Each of the next n lines contains two integers a_i and b_i ($1 \le a_i, b_i \le 10^9$, $a_i \ne b_i$) — the arguments of the *swap* operation.

Output

Print a single integer — the number of inversions in the resulting sequence.

Sample test(s)

input
2 4 2 1 4
output
4

input
3 1 6 3 4 2 5
output
15

Note

In the first sample the sequence is being modified as follows: $\{1, 2, 3, 4, 5, \ldots\} \to \{1, 4, 3, 2, 5, \ldots\} \to \{2, 4, 3, 1, 5 \ldots\}$. It has 4 inversions formed by index pairs $(1, 4)$, $(2, 3)$, $(2, 4)$ and $(3, 4)$.