Task	CETVRTA	PEG	PRINOVA	ZAPIS	SREDNJI	STAZA
Input	standard input (keyboard)					
Output	standard output (screen)					
Memory limit (heap+stack)	32 MB					
Time limit (per test)	1 second					
Number of tests	5	6	10	10	10	10
Points per test	4	5	3	6	7	9
Total points	**20**	**30**	**30**	**60**	**70**	**90**
	300					

Note: The evaluation system has two Intel Pentium 4 3.0 GHz processors and is running the Linux operating system. The following compile options are used for different languages:

- C: -O2 -s -static -std=c99 -lm
- C++: -O2 -s -static -lm
- Pascal: -O1 -XS

1. CETVRTA

Mirko needs to choose four points in the plane so that they form a rectangle with sides parallel to the axes. He has already chosen three points and is confident that he hasn't made a mistake, but is having trouble locating the last point. Help him.

Input

Each of the three points already chosen will be given on a separate line. All coordinates will be integers between 1 and 1000.

Output

Output the coordinates of the fourth vertex of the rectangle.

Sample test data

input

```
5 5
5 7
7 5
```

output

```
7 7
```

input

```
30 20
10 10
10 20
```

output

```
30 10
```

2. PEG

In the famous logic game Peg, pieces jump over other pieces to remove them from the game, until only one piece is left.

Here is the initial layout of the board:

```
  ooo
  ooo
ooooooo
ooo.ooo
ooooooo
  ooo
  ooo
```

The lowercase letter 'o' represents a piece, while the character '.' is an empty square. In one move, a player may choose one piece and one of the four main directions (up, down, left, right), if there is another piece in that direction, and an empty square behind it. The chosen piece jumps over the other piece and settles in the empty square behind it, while the piece being jumped over is removed from the game.

Write a program that calculates the number of legal moves, given the state of the board.

Input

The board is represented by seven lines containing seven characters each. The first two and last two characters on the first two and last two lines are always spaces, and all remaining characters are either 'o' (lowercase letter) or '.' (period character).

Output

Output the number of legal moves.

Sample test data

input

```
  ooo
  ooo
ooooooo
ooo.ooo
ooooooo
  ooo
  ooo
```

output

```
4
```

input

```
  ooo
  ooo
..ooo..
oo...oo
..ooo..
  ooo
  ooo
```

output

```
12
```

Brojko and Brojana are happily married with N little boys. The boys are named with distinct **even** integers $P_1, P_2, ..., P_N$.

Brojko and Brojana are expecting an addition to their family and have to come up with a nice name for the little girl. They have decided that the name will be an **odd** integer in the range [A, B]. Because they find all integers in that range equally beautiful, they have decided to choose the number which maximizes the distance to the name of the closest of the N boys.

More precisely, they seek an odd integer $X \in [A, B]$ such that the expression

$$\min\{|X - P_i|, i \in [1, N]\}$$

is as large as possible.

Write a program that determines the name for the little girl. If there are multiple solutions, output any of them.

Input

The first line contains an integer N ($1 \leq N \leq 100$), the number of boys.

The second line contains N distinct even integers, the names of the boys. The integers will be less than 10^9.

The third line contains the integers A and B ($1 \leq A < B \leq 10^9$), the range of names they are considering for the girl.

Output

Output an integer, the name for the little girl.

Sample test data

input

```
3
2 6 16
20 50
```

output

```
49
```

input

```
3
2 6 16
3 15
```

output

```
11
```

input

```
3
2 6 16
1 7
```

output

```
5
```

A regular bracket-sequence is a string of characters consisting only of opening and closing brackets, and satisfying the following conditions:

- An empty string is a regular bracket-sequence.
- If A is a regular bracket-sequence, then (A), [A] and {A} are also regular bracket-sequences.
- If A and B are regular bracket-sequences, then AB is also a regular bracket-sequence.

For example, the sequences [({})], [](){} i [{}]()[{}] are regular, but the sequences [({{([, []({)} and [{}])([{}] are not.

Ivica has found a string which looks like it could be a regular bracket-sequence. Some of the characters have become smudged and illegible, and could have been any character.

Write a program that calculates how many ways the illegible characters in the string can be replaced by brackets so that the result is a regular bracket-sequence. This number can be very large, so output only its **last 5 digits**.

Input

The first line contains an even integer N ($2 \le N \le 200$), the length of the string.

The second line contains the string. Illegible characters are represented by the '?' character.

Output

Output the number of regular bracket-sequences the string could have read.

Sample test data

input	input	input
6 ()()()	10 (?([?)]?}?	16 ???[???????]????
output	**output**	**output**
1	3	92202

In the second example, the three matching regular bracket-sequences are `({([()]})`, `()([()]{})` and `([([])]{})`.

Consider a sequence A of integers, containing N integers between 1 and N. Each integer appears exactly once in the sequence.

A subsequence of A is a sequence obtained by removing some (possibly none) numbers from the beginning of A, and then from the end of A.

Calculate how many different subsequences of A of **odd** length have their median equal to B. The median of a sequence is the element in the middle of the sequence after it is sorted. For example, the median of the sequence {5, 1, 3} is 3.

Input

The first line contains two integers, N ($1 \leq N \leq 100000$) and B ($1 \leq B \leq N$).

The second line contains N integers separated by spaces, the elements of sequence A.

Output

Output the number of subsequences of A whose median is B.

Sample test data

input

```
5 4
1 2 3 4 5
```

output

```
2
```

input

```
6 3
1 2 4 5 6 3
```

output

```
1
```

input

```
7 4
5 7 2 4 3 1 6
```

output

```
4
```

In the fourth example, the four subsequences of A with median 4 are {4}, {7, 2, 4}, {5, 7, 2, 4, 3} and {5, 7, 2, 4, 3, 1, 6}.

6. STAZA

A bicycle race is being organized in a country. The transport network of the country consists of N cities numbered 1 through N, with M bidirectional roads connecting them. We will use the following terms:

- A **path** is a sequence of roads in which each road starts in the city the preceding road ended in.
- A **simple path** is a path which never visits a city more than once.
- A **ring** is a **simple** path ending in the same city it started in.

The network is such that there is **at least one path** between every pair of cities. Additionally, every **road** in the network is part of **at most one ring**.

Your task is to find the longest path for the race satisfying two constraints:

- The path may begin in any city, but must end in city 1.
- The path may visit a city more than once, but it must not contain any road more than once.

Input

The first line of input contains two integers N and M ($2 \le N \le 10\,000$, $1 \le M \le 2N-2$) – the numbers of cities and roads in the network.

Each of the following M lines contains two different integers A and B ($1 \le A, B \le N$). These numbers indicate that there is a bidirectional road between cities A and B. No two cities will be directly connected by more than one road.

Output

Output the length of the longest race path on a single line.

Sample test data

input

```
4 3
1 2
1 3
2 4
```

output

```
2
```

input

```
6 6
1 2
1 3
2 4
3 4
3 5
5 6
```

output

```
5
```

input

```
5 6
1 2
2 3
3 4
4 5
5 3
3 1
```

output

```
6
```