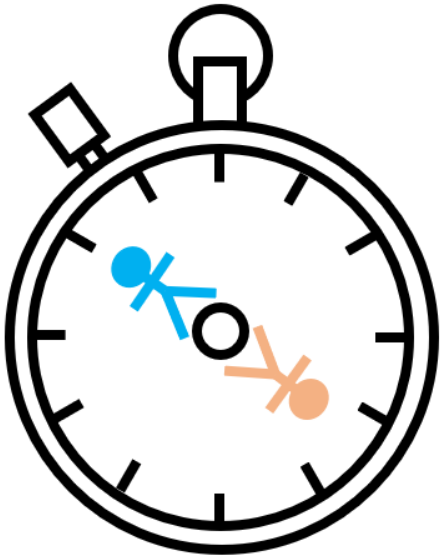
# Problem C - Caçadinhas Cronometradas

Check our instructions page for detailed information on the qualification and the format of this problem.

André and Bruno often play tag in the gardens of the Computer Science Department. The rules are simple: we can imagine the garden as an $N$ by $N$ board, where some cells are occupied (you cannot step on flower beds or climb trees!). Initially, André occupies an unoccupied cell $A$, while Bruno occupies another unoccupied cell $B$. Once the game starts:

- Bruno remains still in his initial position.
- Each second, André moves to an adjacent unoccupied cell (up, left, down, or right). When André occupies cell $B$, where Bruno is located, he can tag Bruno and finish the game.



## Part I

Bruno has plenty of time on his hands... This led him to wonder: given a positive integer $K$, can they play a game of tag that ends but lasts $K$ seconds or more?

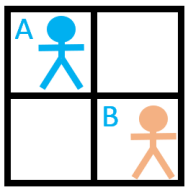
Given the garden's side length, $N$, the initial positions of André, $A$, and Bruno, $B$, and a positive integer $K$, determine if it's possible for the game of tag to end after $K$ **or more** seconds.

André and Bruno will play a total of $T$ independent games, each potentially with different values for $N$, $K$, $A$, and $B$. You must determine the answer for each game.

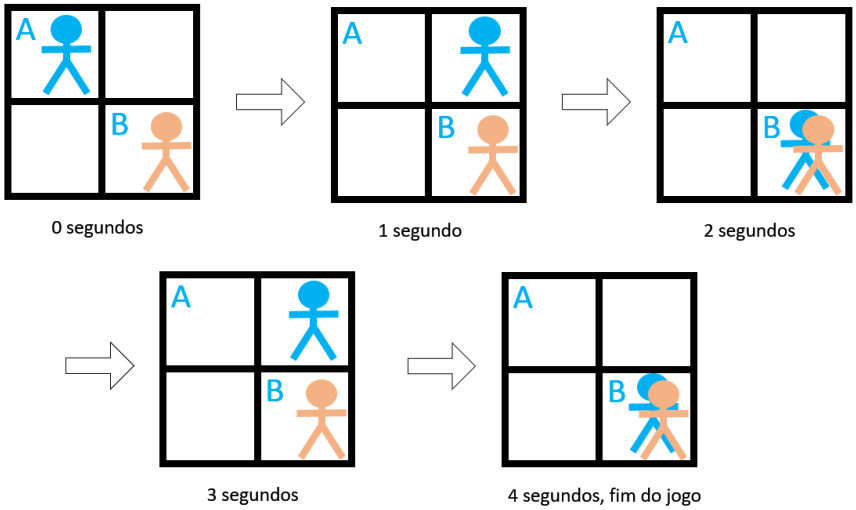
### Example

Consider two possible scenarios.

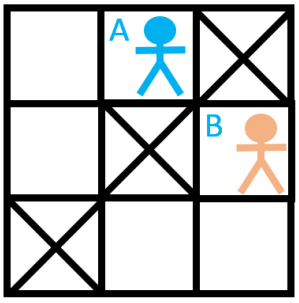
In the following image, André is positioned in the cell at row 1, column 1, and Bruno in the cell at row 2, column 2.




With $\boldsymbol{K} = 3$ and the initial configuration shown above, it is possible for the game to end in 3 or more seconds. Consider André's following sequence of moves, allowing the game to end in $4 \geq 3$ seconds.




Notice that even though André is in Bruno's cell after 2 seconds, he doesn't have to tag him immediately and can wait until 4 seconds. Therefore, the answer to this scenario is YES.




Now consider the above scenario. It's clear that André cannot reach Bruno, regardless of the time available. Thus, the answer to this case would be NAO.

## Constraints

The following limits are guaranteed in all test cases for this Part that will be provided to the program:

| | |
|---|---|
| $1 \leq \boldsymbol{T} \leq 20$ | Number of games |
| $1 \leq \boldsymbol{N} \leq 100$ | Garden side length |
| $1 \leq A_l, A_c, B_l, B_c \leq N$ | Initial coordinates |
| $1 \leq \boldsymbol{K} \leq 10^9$ | Minimum desired game duration, in seconds |

The test cases for this Part of the problem are organized into a single group:

| Subtask | Points | Additional Constraints |
|---|---|---|
| 1 | 20 | No additional restrictions |

# Part II

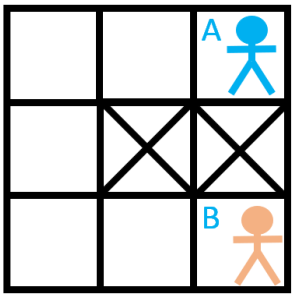
André, despite his fondness for playing tag, must soon leave the gardens. Is there still time for one more game?

Given the garden's side length, $\boldsymbol{N}$, the initial positions of André, $\boldsymbol{A}$, and Bruno, $\boldsymbol{B}$, and a positive integer $\boldsymbol{K}$, determine if it's possible for the game of tag to end within $\boldsymbol{K}$ **or fewer** seconds.

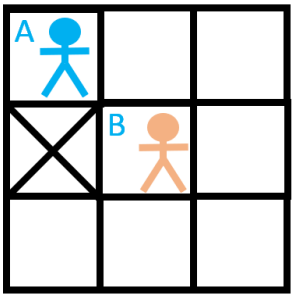
André and Bruno will play a total of $\boldsymbol{T}$ independent games, each potentially with different values for $\boldsymbol{N}$, $\boldsymbol{K}$, $\boldsymbol{A}$, and $\boldsymbol{B}$. You must determine the answer for each game.

## Example

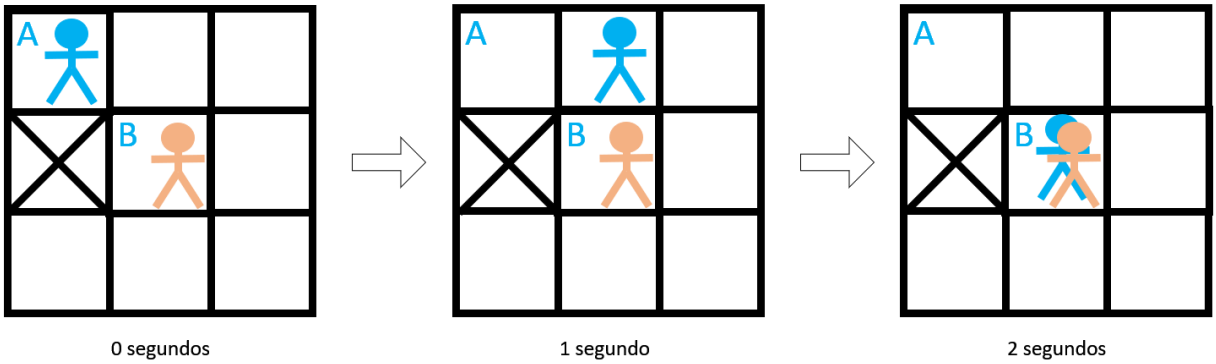
Consider two possible scenarios.




With $\boldsymbol{K} = 4$ and the initial configuration shown above, it's impossible for the game to end in 4 or fewer seconds. Thus, the answer to this case would be NAO.




Now consider the scenario above, with $\boldsymbol{K} = 5$. Observe André's sequence of moves, enabling him to end the game in 5 or fewer seconds.




Thus, the answer to this scenario would be SIM.

## Constraints

The following limits are guaranteed in all test cases for this Part that will be provided to the program:

| | |
|---|---|
| $1 \leq \boldsymbol{T} \leq 20$ | Number of games |
| $1 \leq \boldsymbol{N} \leq 100$ | Garden side length |
| $1 \leq A_l, A_c, B_l, B_c \leq N$ | Initial coordinates |
| $1 \leq \boldsymbol{K} \leq 10^9$ | Maximum desired game duration, in seconds |

The test cases for this Part of the problem are organized into a single group:

| Subtask | Points | Additional Constraints |
|---|---|---|
| 2 | 30 | No additional restrictions |

# Part III

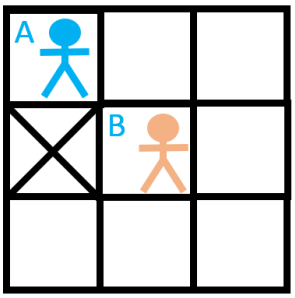
Having acknowledged André's hurry and Bruno's willingness to play, the two friends arrive at a question: can they play a game of tag that lasts exactly $\boldsymbol{K}$ seconds?

Given the garden's side length, $\boldsymbol{N}$, the initial positions of André, $\boldsymbol{A}$, and Bruno, $\boldsymbol{B}$, and a positive integer $\boldsymbol{K}$, determine if it's possible for the game of tag to end in **exactly** $\boldsymbol{K}$ seconds.

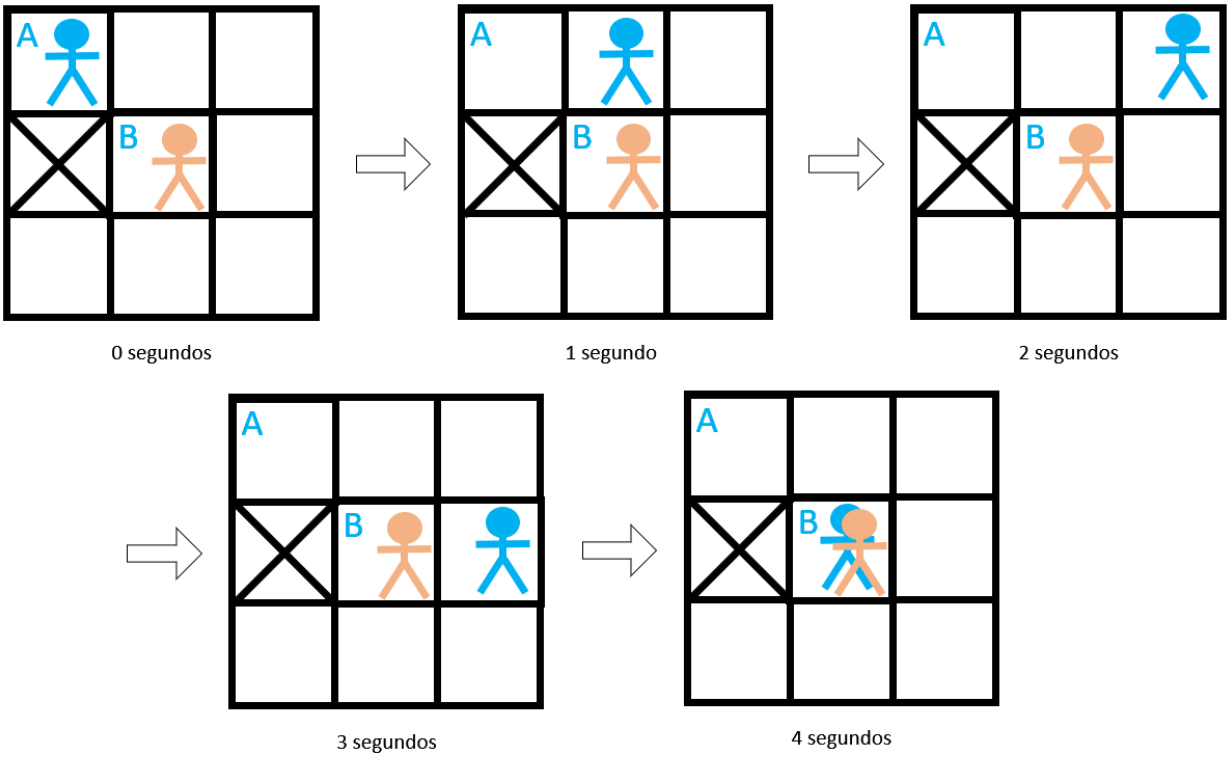
André and Bruno will play a total of $\boldsymbol{T}$ independent games, each potentially with different values for $\boldsymbol{N}$, $\boldsymbol{K}$, $\boldsymbol{A}$, and $\boldsymbol{B}$. You must determine the answer for each game.

## Example

Consider the following scenario.




With $\boldsymbol{K} = 4$ and the initial configuration shown above, it's possible for the game to end exactly after 4 seconds. Just consider the following sequence of moves:




Thus, the answer to this scenario would be SIM.

## Constraints

The following limits are guaranteed in all test cases for this Part that will be provided to the program:

| | |
|---|---|
| $1 \leq \boldsymbol{T} \leq 20$ | Number of games |
| $1 \leq \boldsymbol{N} \leq 100$ | Garden side length |
| $1 \leq A_l, A_c, B_l, B_c \leq N$ | Initial coordinates |
| $1 \leq \boldsymbol{K} \leq 10^9$ | Exact desired game duration, in seconds |

The test cases for this Part of the problem are organized into two groups:

| Subtask | Points | Additional Constraints |
|---|---|---|
| 3 | 20 | $N, K \leq 30$ |
| 4 | 30 | No additional restrictions |

# Input Format

The first line contains an integer $T$, the number of games. Each game is independent and contains:

- The first line has two integers, $N$ and $K$.
- The second line has two integers, $A_l$ and $A_c$, the row and column of cell $A$.
- The third line has two integers, $B_l$ and $B_c$, the row and column of cell $B$.
- The next $N$ lines each contain exactly $N$ characters, each either a `.` (unoccupied) or a `#` (occupied).

It is guaranteed that $A$ and $B$ are distinct and unoccupied.

# Output Format

The output should contain $T$ lines, each with `SIM` or `NAO`, the answer to the respective test case.

**Note:** There should be no space at the end of each line (i.e., after each string, only a newline should appear). If this format is not followed, the result of a submission will be `Presentation Error` (check the instructions for more information).

# Example 1 Input

```
1
2
2 3
1 1
2 2
..
..
3 4
1 2
2 3
..#
.#.
#..
```

## Example 1 Output

```
SIM
NAO
```

## Example 1 Description

This example corresponds to the Part I examples mentioned in the statement.

## Example 2 Input

```
2
2
3 4
1 3
3 3
...
.##
...
3 5
1 1
2 2
...
#..
...
```

## Example 2 Output

```
NAO
SIM
```

## Example 2 Description

This example corresponds to the Part II examples mentioned in the statement.

## Example 3 Input

```
3
1
3 4
1 1
2 2
...
#..
...
```

## Example 3 Output

```
SIM
```

## Example 3 Description

This example corresponds to the Part III example mentioned in the statement.