Gans Media Retro Games

Author: Rengga & Tamam

Timelimit: 1s

Deskripsi Soal

Ren dan Beabadoobee ingin bermain catur malam ini! Namun, setelah bermain terlalu lama, mereka mulai jenuh. Ren pun mendapat ide: “Bisakah aku membuat algoritma yang menentukan apakah kudaku bisa bergerak dari satu kotak ke kotak lainnya, dengan menghindari bidak-bidak yang menghalangi jalannya? ” Bantu Ren membuat algoritma tersebut supaya bisa membuat Beabadoobee kagum dan jatuh cinta!

Format Input

```
N
x1 y1
x2 y2
grid[0][0] grid[0][1] ... grid[0][N-1]
grid[1][0] grid[1][1] ... grid[1][N-1]
...
grid[N-1][0] grid[N-1][1] ... grid[N-1][N-1]
```

1. Baris Pertama adalah bilangan bulat N, yang menunjukkan ukuran Papan catur
2. Baris Kedua adalah bilangan bulat x1 dan y1, yang menunjukkan Posisi awal Kuda
3. Baris Ketiga adalah bilangan bulat x2 dan y2, yang menunjukkan Tujuan Kuda

4. N baris selanjutnya adalah sebuah grid N x N yang dapat berisi 0 atau 1. 0 menandakan petak tersebut kosong, dan 1 menandakan petak tersebut ada bidaknya

Format Output

Cetak YES apabila kuda dapat berpindah dari posisi awal ke tujuan
Cetak NO apabila kuda tidak dapat berpindah dari posisi awal ke tujuan

Batasan

$1 \leq N \leq 100$

Contoh Input 1

```
4
0 0
1 1
0 0 0 1
0 0 1 1
1 1 0 0
0 0 0 0
```

Contoh Output 1

```
NO
```

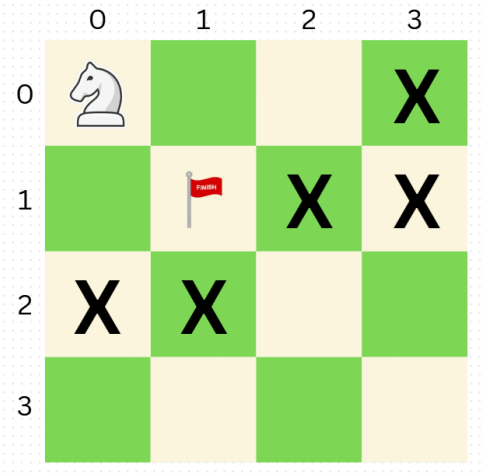
Contoh Input 2

```
8
0 0
3 1
0 0 0 1 0 0 0 1
0 0 1 1 1 1 0 0
1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
1 1 1 0 0 0 1 1
0 0 0 0 0 0 0 0
1 1 1 1 0 0 0 0
0 1 0 1 0 1 0 1
```

Contoh Output 2

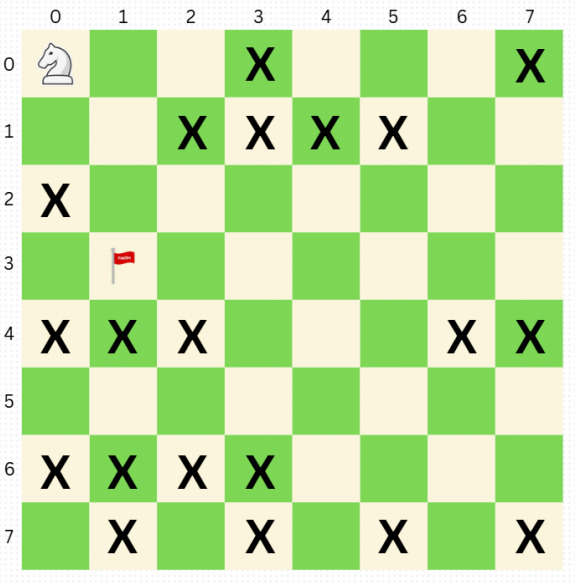
```
YES
```

Penjelasan Contoh 1



Dapat dilihat bahwa Kuda tidak dapat kemanapun, karena terdapat bidak di jalannya. Oleh karena itu, **cetak output NO**

Penjelasan Contoh 2



Salah satu cara kuda untuk menuju tujuan:
(0,0) -> (2,1) -> (3,3) -> (5,2) -> (3,1)

Karena terdapat jalan untuk menuju tujuan, **maka cetak output YES**

Gans Media Retro Games

Author: Rengga & Tamam

Timelimit: 1s

Problem Description

Ren and Beabadoobee want to play chess tonight! However, after playing for too long, they start to get bored. Ren then comes up with an idea: "Can I create an algorithm to determine if my knight can move from its starting square to a target square, while avoiding the pieces blocking its path?" Help Ren create this algorithm to impress Beabadoobee and win her heart!

Input Format

```
N
x1 y1
x2 y2
grid[0][0] grid[0][1] ... grid[0][N-1]
grid[1][0] grid[1][1] ... grid[1][N-1]
...
grid[N-1][0] grid[N-1][1] ... grid[N-1][N-1]
```

1. The first line contains an integer N, representing the size of the chessboard (N×N).
2. The second line contains two integers x1 and y1 , representing the knight's starting position.
3. The third line contains two integers x 2 and y 2 , representing the knight's target position
4. The next N lines represent an N×N grid containing 0 or 1. A 0 indicates that the square is empty, and a 1 indicates that the square is occupied by an obstacle (a chess piece).

Output Format

Print YES if the knight can move from the starting position to the target position.
Print NO if the knight cannot move from the starting position to the target position.

Constraint

$1 \leq N \leq 100$

Input Example 1

```
4
0 0
1 1
0 0 0 1
0 0 1 1
1 1 0 0
0 0 0 0
```

Output Example 1

```
NO
```

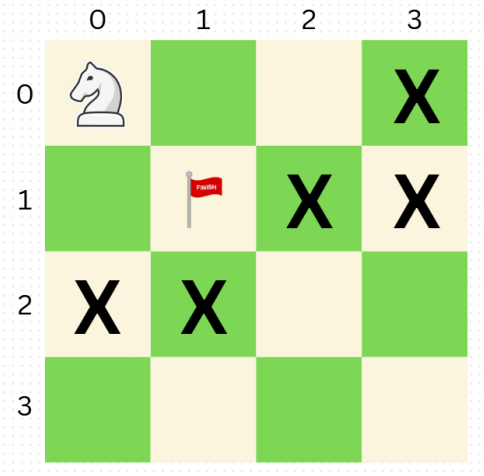
Input Example 1

```
8
0 0
3 1
0 0 0 1 0 0 0 1
0 0 1 1 1 1 0 0
1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
1 1 1 0 0 0 1 1
0 0 0 0 0 0 0 0
1 1 1 1 0 0 0 0
0 1 0 1 0 1 0 1
```

Output Example 1

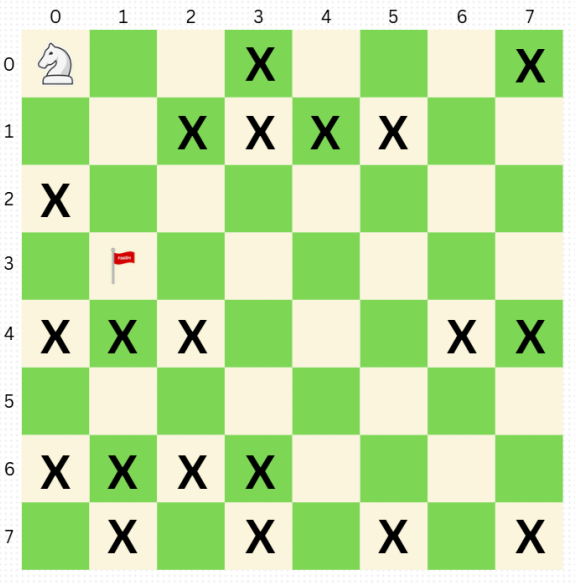
```
YES
```

Explanation 1



As you can see, the knight cannot go anywhere because the path is blocked. So he can't go to the target position. Therefore, **you should print NO**

Explanation2



One of the ways for the knight to go to the target;
(0,0) -> (2,1) -> (3,3) -> (5,2) -> (3,1)

Because there is a possible way for the knight to go to the target, **print YES**