Is there a target?

DiPS CodeJam 23

Prompt

Given a matrix of size at least 3 × 3 where every element $e \in \mathbb{N}$, determine if it contains at least one "target" pattern, defined as:

$$\begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & n & n & n & * \\ * & * & n & - & n & * \\ * & * & n & n & n & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \end{bmatrix}$$

where n is a constant, $-$ is any number other than n, and $*$ is any number.

Input Format

- The first line contains an integer n, denoting the size of the matrix.
- The next n lines contain n space-separated numbers each.

Output Format

Your output must contain "true" if a target pattern exists, and "false" if it does not.

Sample Input/Output

Input	Output

6
6 7 4 5 5 6
7 6 2 1 5 9
7 7 8 7 4 1
3 4 6 8 6 8
2 5 6 9 8 2
8 2 2 7 6 2
	false

Sample Program

```
n=int(input())

matrix=[]

for i in range(n):
  matrix.append(list(map(int, input().strip().split())))

def safeCell(i, j):
```

```python
  try:
    assert i>-1 and j>-1 # prevent back-indexes
    return matrix[i][j]
  except:
    return None

def checkEquality(arr):
  if None in arr:
    return False
  else:
    if len(set(arr))==1:
      return True
  return False

def isThereATarget():
  for i in range(n):
    for j in range(n):
      bounding_cells = [
        safeCell(i-1, j-1),
        safeCell(i-1, j),
        safeCell(i-1, j+1),
        safeCell(i, j-1),
        safeCell(i, j+1),
        safeCell(i+1, j-1),
        safeCell(i+1, j),
        safeCell(i+1, j+1),
      ]

      if checkEquality(bounding_cells) and bounding_cells[0]!=matrix[i][j]:
        return True
  return False

print("true" if isThereATarget() else "false")
```