

## A cop and a robber

A cop is trying to catch a robber moving along the edges of a graph. The two players play alternatingly.

1. The game starts by the cop choosing his initial vertex.
2. The robber then chooses his starting vertex.
3. The cop's move consists of him moving to a neighbouring vertex (i.e. a vertex that is connected by an edge to the current vertex of the cop) or waiting (i.e. not moving).
4. The robber's move consists of him moving to a neighbouring vertex. Note that he is not allowed to wait.
5. The robber's move consists of him moving to a neighbouring vertex. Note that he is not allowed to wait.
  - (a) both players are on the same vertex after a move of either player — cop captures the robber so wins;
  - (b) the position repeats (any position is defined by cop's vertex  $c$ , robber's vertex  $r$  and the side  $T$  whose turn it is to move next; a position repeats if there has been an earlier move after which all three parameters  $c$ ,  $r$  and  $T$  were the same) — this corresponds to the robber being able to avoid the cop indefinitely, so robber wins.

## Task

Given the graph, determine which player wins if both play optimally. Moreover, play the game for the winning player and win.

## Implementation

You need to implement the function `get_cake(R, C, M)` which takes the following parameters:

- $R$  — the number of rows in the map  $R$ ,
- $C$  — the number of columns in the map  $C$ ,
- $M$  — a two-dimensional array, where  $M[i][j]$  ( $0 \leq i < R$ ,  $0 \leq j < C$ ) is the cell in the  $i$ -th row and  $j$ -th column of the map.

## Example

## Scoring



**Subtask 1 (25 points):**

**Subtask 2 (25 points):**

**Subtask 3 (25 points):**

**Subtask 4 (25 points):**

## Constraints

**Time limit:** 1 s.

**Memory limit:** 256 MB.

Need to  
add info  
about  
graders.