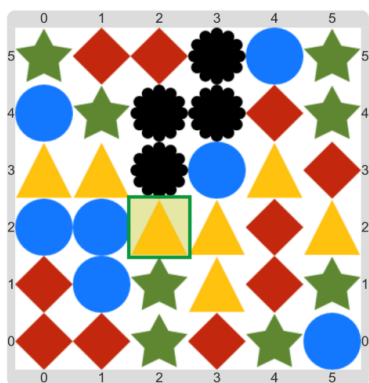


El Shariki (las esferas, en ruso) es un videojuego de tipo puzzle desarrollado originalmente para DOS por Eugene Almezhin. El objetivo del juego era intercambiar una esfera por otra lindante (Pueden establecerse otros movimientos especiales, además de este), de forma tal de obtener líneas horizontales o verticales que contuviera tres o más esferas del mismo color. Al lograrlo, esas esferas explotaban, otorgando puntos al jugador, y apareciendo en su lugar nuevas esferas de colores aleatorios. El juego se terminaba cuando no había más movimientos válidos para realizar. Shariki se volvió muy popular y su mecánica vió múltiples reencarnaciones bajo distintos nombres, como Bejeweled, Jewel Quest, Candy Crush Saga y Pokemon Shuffle.

El juego se desarrolla en una grilla rectangular de tamaño variable, en donde en cada ubicación puede haber, o bien una ficha, o bien un obstáculo. Las fichas pueden ser de 4 tipos distintos: estrellas, círculos, rombos o triángulos. El juego consiste en intercambiar una ficha por otra, mediante un movimiento válido. Un movimiento es válido si:

- Los obstáculos consisten en elementos que se ubican en los mismos lugares que podrían ocupar fichas, pero no pueden moverse, limitando la cantidad de movimientos que se pueden realizar, y aumentando el nivel de dificultad del juego.

A continuación hay un tablero de ejemplo que muestra un posible tablero inicial, con vestimentas:



Hay dos movimientos válidos posibles, o bien moviendo el triángulo en (4,3) hacia el Sur, o bien moviendo el rombo en (5, 3) hacia el Oeste. En ambos casos se formarán líneas.

Ejercicio #01) Definir **formaDefichaActual** que debe respetar lo siguiente:

function formaDefichaActual

Propósito.

Describe el número que se corresponde a la forma de la ficha de la ubicación actual.

Precondición.

El cabezal se encuentra sobre una ficha.

Tipo.

Número.

Ejercicio #02) Se pide desarrolle la función **haySucesiónAI_** que indica si hay una sucesión de al menos 3 fichas similares hacia la dirección dada, comenzando por la ficha en la celda actual. Puede asumir que hay al menos 2 celdas en la dirección dada.

Para solucionar este problema puede hacer uso de la siguiente primitiva:

function formaDeFichaA_AI_(distanciaAMirar, dirección)

Propósito.

Describe la forma de la ficha (número) que se encuentra tantos lugares hacia **direcciónAMirar** como **distanciaAMirar**. Describe 0 si no hay una ficha en dicha celda.

Precondición.

Hay una celda a **distanciaAMirar** en dirección **direcciónAMirar**.

Parámetros.

distanciaAMirar. Número. La distancia a la cual mirar la forma de la ficha.

dirección. Dirección. La dirección hacia la cual mirar la forma de la ficha.

Tipo.

Número.