

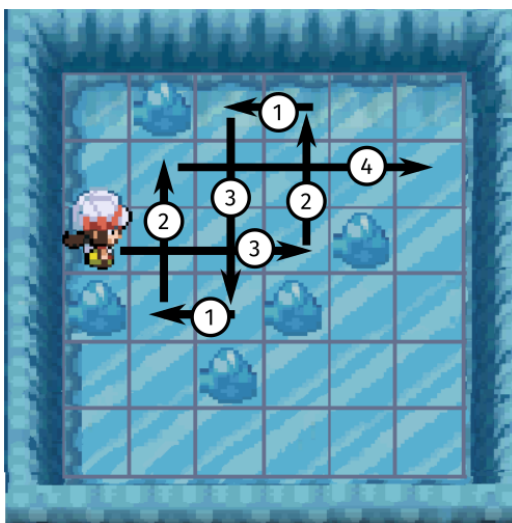
## Problema La cueva de hielo

Terry es una chica a la que le gustan los desafíos. En este momento ella se encuentra participando en una difícil competencia organizada por la Organización Cultural de Internet (OCI). Esta competencia consiste en completar complejos laberintos de la cueva de hielo de cierta franquicia de criaturas de bolsillo.

Un laberinto consiste en una grilla de  $N \times N$  casillas. Cada casilla es identificada por sus coordenadas  $x$  e  $y$  ( $0 \leq x, y \leq N - 1$ ). Algunas casillas en el laberinto están vacías mientras que otras pueden contener una roca. Además dos casillas especiales son identificadas como la casilla inicial y final. Estas casillas están siempre vacías, es decir, no contienen rocas.

El objetivo es mover a un personaje por el laberinto llevándolo desde la casilla inicial hasta la final. Cada vez que el personaje se mueve a una casilla contigua tarda una unidad de tiempo. Lo complejo de los laberintos es que el piso es de hielo y por lo tanto el personaje no puede moverse libremente. Cada vez que el personaje comienza a moverse en una dirección debe continuar moviéndose en esa dirección hasta chocar con una roca o con el borde del laberinto. Notar que no basta con que el personaje pase por sobre la casilla final, este tiene que quedar detenido en esa posición.

La figura de más abajo muestra un laberinto y un posible camino. El personaje comienza en la casilla  $(0, 3)$  y el objetivo es llevarlo a la casilla  $(5, 4)$ . En primer lugar, el personaje se mueve hacia la derecha hasta chocar con la piedra en la casilla  $(4, 3)$  quedando detenido en la casilla  $(3, 3)$ . Este movimiento tarda 3 unidades de tiempo. A continuación se mueve hacia arriba hasta chocar con el borde superior quedando detenido en la casilla  $(3, 5)$  y tardando 2 unidades de tiempo. Los siguientes movimientos llevan al personaje a las casillas  $(2, 5)$ ,  $(2, 2)$ ,  $(1, 2)$ ,  $(1, 4)$  y finalmente a la casilla  $(5, 4)$ . El tiempo total ocupado en estos movimientos es  $3 + 2 + 1 + 3 + 1 + 2 + 4 = 16$ .



Terry es una chica muy hábil, pero siempre está ocupada organizando eventos y esta vez no ha tenido tiempo de entrenar para resolver los laberintos. Terry siempre está disponible para ayudar a quién lo necesite. Ahora es momento de que tú la ayudes resolviendo estos laberintos.

### Subtarea 1 (50pts)

La primera subtarea consiste en determinar si es posible llevar al personaje desde la casilla inicial a la final. En esta subtarea no es importante el tiempo que toma mover al personaje.

- `boolean es_posible(bool[] [] rocas, int N, int xi, int yi, int xf, int yf):`

Esta función debe determinar si es posible para el laberinto especificado llevar al personaje desde la posición inicial a la final. Si es posible la función debe retornar `true`, en caso contrario debe retornar `false`.

- `rocas`: Arreglo de dos dimensiones especificando las casillas del laberinto que contienen una roca. Una posición con `false` corresponde a una casilla vacía. Una posición con `true` corresponde a una casilla con una roca.
- `N`: Tamaño de la grilla.
- `xi`: Coordenada  $x$  de la posición inicial.
- `yi`: Coordenada  $y$  de la posición inicial.
- `xf`: Coordenada  $x$  de la posición final.
- `yf`: Coordenada  $y$  de la posición final.

### Subtarea 2 (50pts)

En esta subtarea debe encontrarse el mínimo tiempo en que es posible llevar al personaje desde la posición inicial a la final.

- `int minimo(bool[] [] rocas, int N, int xi, int yi, int xf, int yf):`

Esta función debe retornar un entero correspondiente a la mínima cantidad de tiempo en que es posible llevar al personaje desde la posición inicial a la final. En caso de no ser posible la función debe retornar `-1`. Los parámetros de la función son los mismos que los de `es_posible`.

### Grader de Ejemplo