




Mini Tetris

Como usted debe saber, el popular juego de computador “Tetris” fue inventado por el programador ruso Alexey Pajitnov. En este problema usted necesita imprimir un programa que juegue una versión simplificada de este juego.

El campo de juego es un rectángulo vertical, llamado el “pozo”. Aparecen figuras aleatorias en la parte superior del pozo, el jugador elige la posición horizontal y la rotación de la figura, después la figura cae en el pozo. El objetivo del juego es crear una línea horizontal llena sin interrupciones. Cuando se crea una línea así, desaparece, y caen todos los bloques encima de la línea borrada caen.

En esta modificación del juego, el tamaño del pozo es 3×4 unidades, y únicamente hay tres tipos de figuras:

Tipo	Figura
1	
2	
3	

Usted pierde si en algún punto del juego hay cinco líneas no vacías en el pozo. Usted gana si usted no perdió después que han caído n figuras.

Usted necesita escribir un programa que juegue el juego antes descrito y gane independientemente de que figuras aparezcan.

Detalles de implementación

Usted debe implementar cuatro funciones (métodos):

- `void init(int n)`. Esta función se llama antes de cualquier otra función.
- `void new_figure(int figure_type)`. Esta función se llama cuando aparece una nueva figura. `figure_type` es el número de **1** a **3**, indicando el tipo de figura de la tabla antes dada.
- `int getPosition()`. Esta función debe devolver el número de (0) to **2**, la posición

del bloque más a la izquierda de la última figura.

- `int getRotation()`. Esta función debe devolver un número de 0 a 3, el número de rotación en sentido antihorario de la figura.

Las funciones `getPosition` y `getRotation` serán llamadas únicamente después de `new_figure`.

Por favor mire los archivos modelo de implementación proporcionados por detalles en su lenguaje de programación.

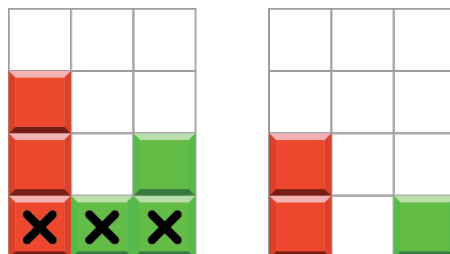
Ejemplo

El calificador hace las siguientes llamadas de funciones:

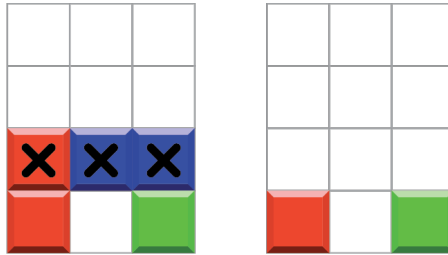
- `init(3)`. Habrán tres figuras.
- `new_figure(1)`. Una figura del tipo 1 cae de la parte superior del pozo.
- `getPosition()` devuelve 0. Esto quiere decir que el jugador quiere poner la figura en la columna más a la izquierda;
- `getRotation()` devuelve 1 (o 3). Esto quiere decir que el jugador quiere rotar la figura verticalmente.
- Después que la figura ha caído, el pozo se ve como esto.



- `new_figure(2)`.
- `getPosition()` devuelve 1.
- `getRotation()` devuelve 1.
- Después que la figura ha caído, la primera línea está llena, por lo tanto desaparece y se ve como esto.



- `new_figure(1)`.
- `getPosition()` devuelve 1.
- `getRotation()` devuelve 0 (o 2).
- Después que la figura ha caído, la segunda línea está llena, por lo tanto el pozo se ve como esto.



Subtareas

En todas las subtareas $n \leq 1000$.

1. (7 puntos) Todas las figuras son del tipo **1**,
2. (13 puntos) Todas las figuras son del tipo **2**,
3. (21 puntos) Todas las figuras son del tipo **3**,
4. (53 puntos) Las figuras pueden ser de diferentes tipos.

Calificador Ejemplo

El calificador ejemplo lee la entrada en el siguiente formato:

- línea 1: Un entero n .
- línea 2: n enteros: tipos de figuras.

Notas del Lenguaje

Por favor, use los archivos modelos proporcionados para detalles de implementación en su lenguaje de programación.