




Mini Tetris

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

El campo de juego es un rectángulo vertical, llamado el “pozo”. Aparecen por arriba del pozo aleatoriamente figuras compuestas de bloques de una unidad cuadrada, el jugador escoge la posición horizontal y la rotación de la figura, después de eso la figura cae en el pozo. El objetivo del juego es crear líneas horizontales sin interrupciones. Cuando una línea horizontal completa es creada entonces desaparece y todos los bloques arriba de la línea quitada caen.

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

Tipo	Figura
1	
2	
3	

Usted pierde si en algún momento hay cinco líneas no vacías en el pozo. Usted gana si no pierde después de caer n figuras al pozo.

Usted necesita escribir un programa que juegue esta versión descrita arriba y gane sin importar cuales figuras aparezcan.

Detalles de Implementación

Usted debe implementar cuatro funciones (métodos):

- `void init(int n)`. Esta función es llamada antes que cualquier otra función.
- `void new_figure(int figure_type)`. Esta función es llamada cuando una nueva figura aparece. `figure_type` es un número de 1 a 3, indicando el tipo de figura de la tabla anterior.

- `int getPosition()`. Esta función debe retornar un número entre `0` y `2`, la posición del bloque más a la izquierda de la última figura.
- `int getRotation()`. Esta función debe retornar un número entre `0` y `3`, el número de rotaciones en sentido antihorario de la figura.

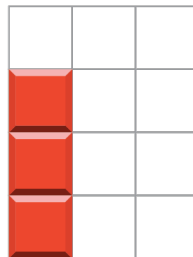
Las funciones `getPosition` y `getRotation` solo serán invocadas después de ser invocada `new_figure`.

Por favor, use las plantillas proporcionadas para los detalles de la implementación de su lenguaje de programación.

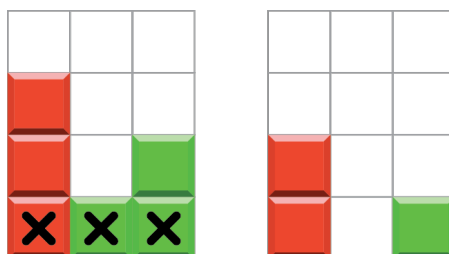
Ejemplo

El calificador realiza la siguiente secuencia de invocaciones:

- `init(3)`. Existirá un total de tres figuras.
- `new_figure(1)`. Una figura del tipo `1` cae de la parte superior del pozo.
- `getPosition()` retorna `0`. Esto significa que el jugador quiere poner la figura en la columna más a la izquierda;
- `getRotation()` retorna `1` (o `3`). Esto quiere decir que el jugador quiere rotar la figura verticalmente.
- Después que la figura cae, el pozo se ve así:



- `new_figure(3)`.
- `getPosition()` retorna `1`.
- `getRotation()` retorna `1`.
- Después que esta figura cae, la primera línea está completa, entonces la misma desaparece y el pozo se ve así.



- `new_figure(2)`.
- `getPosition()` retorna `1`.
- `getRotation()` retorna `0` (o `2`).
- Después que esta figura cae, la segunda línea está completa, entonces desaparece y el pozo se ve así.



Subtareas

En todas las subtareas $n \leq 1000$.

1. (7 puntos) Todas las figuras son del tipo 1.
2. (8 puntos) Todas las figuras son del tipo 2.
3. (10 puntos) Todas las figuras son del tipo 1 o 2.
4. (21 puntos) Todas las figuras son del tipo 3.
5. (54 puntos) Las figuras pueden ser de cualquier tipo.

Calificador de ejemplo

El calificador de ejemplo lee la entrada en el siguiente formato:

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

Notas del lenguaje

Por favor, use las plantillas proporcionadas para los detalles de la implementación de su lenguaje de programación.