Mini Tetris

Popularna gra komputerowa "Tetris" została wymyślona przez rosyjskiego programistę Alexey'a Pajitnova. W tym problemie, musisz napisać program, który będzie grał w uproszczoną wersję tej gry.

Plansza go gry jest prostokątem, zwanym "studnią". Losowe figury złożone z kwadratowych klocków (o rozmiarze jednostkowym) pojawiają się na górze studni, a gracz decyduje o pozycji tej figury w poziomie, a także o jej obrocie. Po tym figura spada na dół studni. Celem gry jest tworzenie poziomych linii wypełnionych klockami bez dziur. Kiedy taka linia zostaje stworzona, znika ona ze studni, a wszystkie klocki powyżej tej linii spadają w dół.

W tej modyfikacji gry, studnia ma rozmiar 3×4 jednostki, i mamy jedynie trzy rodzaje figur:

Typ	Figura
1	
2	
3	

Gra jest przegrana, jeżeli w pewnym momencie w studni znajduje się pięć niepustych linii. Gra jest wygrana, jeżeli nie jest przegrana, po spadku n figur.

Twoim zadaniem jest napisanie programu, który gra w opisaną powyżej grę i wygrywa niezależnie od figur, które pojawią się w trakcie gry.

Szczegóły implementacyjne

Powinieneś zaimplementować następujące cztery funkcje (metody):

- `void init(int n)`. Ta funkcja jest wywoływana przed wywołaniem każdej innej funkcji.
- `void new_figure(int figure_type)`. Ta funkcja wywoływana jest, kiedy pojawia się nowa figura. `figure_type` jest liczbą od 1 do 3, wskazującą typ figury,

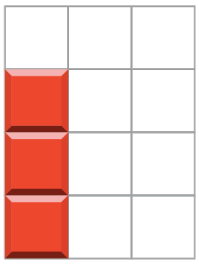
zgodnie z tabelą powyżej.
- `int getPosition()`. Ta funkcja powinna zwrócić liczbę od 0 do 2, pozycję pierwszego od lewej strony klocka ostatniej figury.
- `int getRotation()`. Ta funkcja powinna zwrócić liczbę od 0 do 3, liczbę obrotów figury przeciwnie do ruchu wskazówek zegara.

Funkcje `getPosition` i `getRotation` będą wywoływane jedynie po wywołaniu `new_figure`.

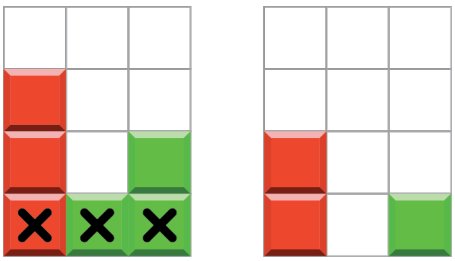
W załączonym przykładowych plikach możesz sprawdzić szczegóły implementacyjne w Twoim języku programowania.

Przykład

Program sprawdzający wywołuje kolejno funkcje z następującymi parametrami:
- `init(3)`. Pojawią się trzy figury.
- `new_figure(1)`. Figura typu 1 spada z góry studni.
- `getPosition()` zwraca 0. To znaczy, że gracz chce ustawić klocek w pierwszej kolumnie od lewej strony.
- `getRotation()` zwraca 1 (lub 3). To znaczy, że gracz chce ustawić figurę pionowo.
- Po tym, jak ta figura upadła, plansza wygląda następująco:

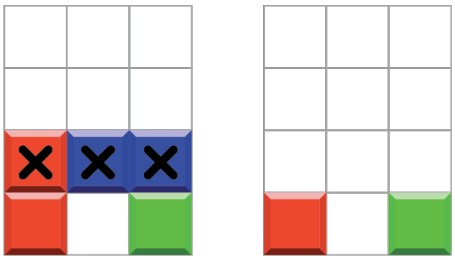


- `new_figure(2)`.
- `getPosition()` zwraca 1.
- `getRotation()` zwraca 1.
- Po tym jak tak figura upadnie, pierwsza od dołu linia jest wypełniona, zatem znika, a cała studnia wygląda następująco:



- `new_figure(1)`.
- `getPosition()` zwraca 1.
- `getRotation()` zwraca 0 (lub 2).
- Po tym, jak upada ostatnia figura, druga linia jest wypełniona, zatem znika, a

studnia wygląda następująco:



Podzadania

We wszystkich podzadaniach $n \leq 1000$.

1. (7 punktów) Wszystkie figury mają typ 1,
2. (13 punktów) Wszystkie figury mają typ 2,
3. (21 punktów) Wszystkie figury mają typ 3,
4. (53 punkty) Figury mogą mieć różne typy.

Przykładowy program sprawdzający

Przykładowy program sprawdzający wczytuje dane w następującym formacie:

- wiersz 1: liczba całkowita n.
- wiersz 2: n liczb całkowitych: typy figur.

Języki programowania

Szczegóły implementacji w Twoim języku programowania znajdują się w dostarczonych plikach z szablonami.