




## Mini Tetris

Kao što vam je vjerovatno poznato, popularnu računarsku igru “Tetris” osmislio je ruski programer Aleksej Pajitnov. U ovom zadatku treba da napišete program koji igra pojednostavljenu verziju ove igre.

Polje za igru je organizovano u obliku pravougaonog vertikalnog okna koje obično zovemo “bunar”. Slučajne figure sastavljene od blokova kvadratnog oblika pojavljuju se na vrhu bunara. Igrač bira horizontalnu poziciju i rotaciju figure, a zatim figura pada na dno bunara. Cilj igre je da se napravi horizontalna linija u kojoj nema praznina. Kada se napravi takva linija, ona nestaje (briše se) a svi blokovi iznad izbrisane linije padaju.

U pojednostavljenoj verziji ove igre, bunar je dimenzija  $3 \times 4$  bloka i postoje samo tri tipa figura:

Tip	Figura
1	
2	
3	

Gubite igru ako u nekom trenutku postoji 5 nepraznih linija u bunaru. Pobjeđujete ako niste izgubili poslije pojavljivanja  $n$  figura na vrhu bunara.

Napišite program koji igra po datim pravilima i pobjeđuje bez obzira koje se figure pojavljuju.

### Detalji implementacije

Potrebno je implementirati četiri funkcije (metoda):

- `void init(int n)`. Ova se funkcija poziva prije svih ostalih funkcija.
- `void new_figure(int figure_type)`. Ova se funkcija poziva kada se pojavljuje nova figura. `figure_type` je broj između 1 i 3, i označava tip figure iz gornje tabele.
- `int getPosition()`. Ova funkcija vraća broj od 0 do 2, poziciju krajnjeg lijevog

bloka posljednje figure koja se pojavila.

- `int getRotation()`. Ova funkcija vraća broj od `0` od `3`, broj rotacija figure u smjeru suprotnom kretanju kazaljke sata.

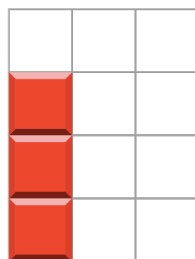
Funkcije `getPosition` i `getRotation` biće pozivane samo poslije poziva `new_figure`.

Koristite date template-fajlove za detalje o implementaciji u izabranom programskom jeziku.

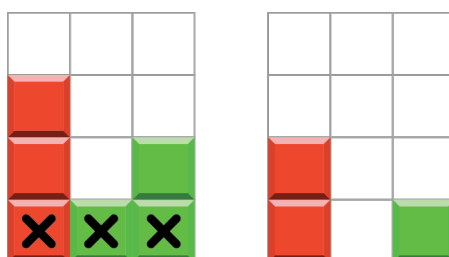
## Primjer

Sistem za ocjenjivanje koristi sljedeće pozive funkcija:

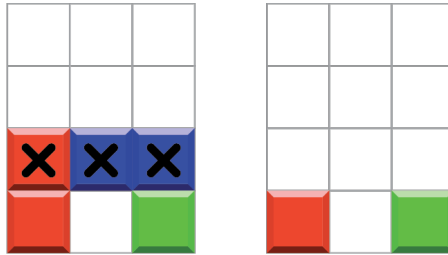
- `init(3)`. Pojaviće se tri figure.
- `new_figure(1)`. Figura tipa `1` pada sa vrha bunara.
- `getPosition()` vraća `0`. To znači da igrač postavlja figuru u krajnju lijevu kolonu;
- `getRotation()` vraća `1` (ili `3`). Igrač želi da rotira figuru vertikalno.
- Poslije pada figure, izgled bunara je sljedeći.



- `new_figure(2)`.
- `getPosition()` vraća `1`.
- `getRotation()` vraća `1`.
- Poslije pada figure, prva linija nema praznina pa nestaje, a bunar ima sljedeći izgled.



- `new_figure(1)`.
- `getPosition()` vraća `1`.
- `getRotation()` vraća `0` (or `2`).
- Poslije pada figure, druga linija je popunjena pa nestaje, tako da bunar ima sljedeći izgled.



## Podzadaci

U svim podzadacima važi  $n \leq 1000$ .

1. (7 bodova) Sve figure imaju tip **1**,
2. (13 bodova) Sve figure imaju tip **2**,
3. (21 bod) Sve figure imaju tip **3**,
4. (53 boda) Figure mogu imati različite tipove.

## Sample Grader

Sistem za ocjenjivanje učitava ulaz u sljedećem formatu:

- red 1: cio broj  $n$ .
- red 2:  $n$  cijelih brojeva: tipovi figura.

## Napomena o programskim jezicima

Koristite date templejt-fajlove za detalje o implementaciji u izabranom programskom jeziku.