Mini Tetris

As you may know, the popular computer game “Tetris” was invented by russian programmer Alexey Pajitnov. In this problem you need to write programm which plays the simplified version of this game.

The playing field is a rectangular vertical shaft, called the “well”. Random figures of unit square blocks appear on top of the wall, the player chooses the horisontal position and rotation of the figure, after that the figure falls down in the well. The objective of the game is to create a horizontal lines filled without gaps. When such a line is created, it disappears, and any blocks above the deleted line fall.

In this modification of the game, the well size is 3×4 units, and there are only three types of figures:

Type	Figure
1	
2	
3	

You lose if at some point there are five non-empty lines of the well. You win if you didn't lose after n figures have fallen.

You need to write program which plays the game described above and wins regardless which figures will appear.

Implementation details

You should implement four functions (methods):

- `void init(int n)`. This function is called before any other function.
- `void new_figure(int figure_type)`. This function is called when the new figure appears. `figure_type` is the number from 1 to 3, indicating the figure type from the table above.
- `int getPosition()`. This function should return the number from 0 to 2, the

position of the leftmost block of last figure.

- `int getRotation()`. This function should return the number from 0 to 3, the number of couter-clockwise rotation of the figure.

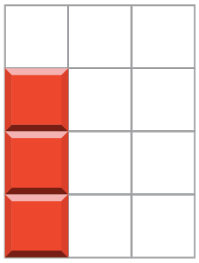
Functions `getPosition` and `getRotation` will be called only after `new_figure`.

Please use the provided template files for details of implementation in your programming language.

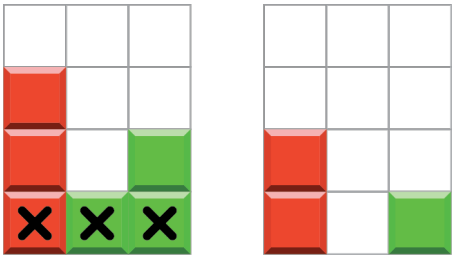
Example

The grader makes the following function calls:

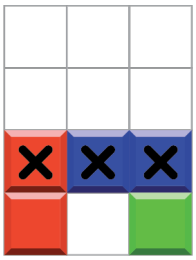
- `init(3)`. There will be three figures.
- `new_figure(1)`. The figure of type 1 falls from the top of the well.
- `getPosition()` returns 0. This means that player wants to put the figure in the leftmost column;
- `getRotation()` returns 1 (or 3). This means that player wants to rotate the figure vertically.
- After the figure has fallen, the well looks like this.

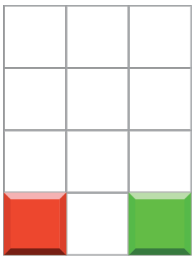


- `new_figure(2)`.
- `getPosition()` returns 1.
- `getRotation()` returns 1.
- After the figure has fallen, the first line is full, so it disappears and well looks like this.



- `new_figure(1)`.
- `getPosition()` returns 1.
- `getRotation()` returns 0 (or 2).
- After the figure has fallen, the second line is full, so it disappears and well looks like this.





Subtasks

In all subtasks $n \leq 1000$.

1. (7 points) All figures have type 1,
2. (13 points) All figures have type 2,
3. (21 points) All figures have type 3,
4. (53 points) Figures can have different types.

Sample Grader

The sample grader reads the input in the following format:

- line 1: One integer n.
- line 2: n integers: types of figures.

Language Notes

Please use the provided template files for details of implementation in your programming language.