Μίνι Τέτρις

Το δημοφιλές παιχνίδι “Tetris” δημιουργήθηκε από τον Ρώσσο προγραμματιστή Alexey Pajitnov. Σε αυτό το πρόβλημα πρέπει να υλοποιήστε μια απλουστευμένη μορφή του παιχνιδιού.

Στο κανόνικό παιχνίδι, διάφορα κομμάτια αποτελούμενα από τετράγωνα μπλοκ ενωμένα μεταξύ τους, πέφτουν μέσα σε ένα "πηγάδι". Τα κομμάτια μπορούν να περιστραφούν καθώς πέφτουν προς τα κάτω. Στόχος του παιχνιδιού, είναι να δημιουργηθούν γραμμές, χωρίς κενά, ενώνοντας τα τετράγωνα μπλοκ που αποτελούν τα κομμάτια. Όταν δημιουργηθεί μια τέτοια γραμμή, αυτή εξαφανίζεται και τα κομμάτια που βρίσκονται από πάνω της πέφτουν.

Σε αυτή την έκδοση του παιχνιδιού το "πηγάδι" είναι μεγέθους 3×4 και υπάρχουν μόνο τρία είδη κομματιών:

Τύπος	Κομμάτι
1	
2	
3	

Χάνετε, όταν δημιουργηθούν πέντε μη-κενές γραμμές στο πηγάδι. Κερδίζετε, όταν δεν χάσετε αφ' ότου πέσουν n κομμάτια.

Να δημιουργήσετε πρόγραμμα που να παίζει το παιχνίδι που περιγράφηκε πιο πάνω και να κερδίζει, ανεξαρτήτως της σειράς εμφάνισης των κομματιών.

Λεπτομέρειες υλοποίησης

Να υλοποιήσετε τέσσερις συναρτήσεις (μεθόδους):

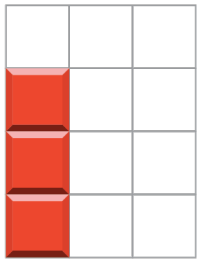
- `void init(int n)` - Αυτή η συνάρτηση θα καλείται πριν από κάθε άλλη συνάρτηση.
- `void new_figure(int figure_type)` - Αυτή η συνάρτηση καλείται όταν εμφανίζεται ένα νέο κομμάτι. `figure_type` ένας ακέραιος αριθμός μεταξύ 0

και 2 συμπεριλαμβανομένων, που υποδεικνύει τον τύπο του κομματιού.
- `int getPosition()`. Αυτή η συνάρτηση πρέπει να επιστρέφει έναν ακέραιο αριθμό μεταξύ 0 και 2 συμπεριλαμβανομένων, τη θέση του αριστερότερου τετραγώνου του τελευταίου κομματιού.
- `int getRotation()`. - Αυτή η συνάρτηση πρέπει να επιστρέφει έναν ακέραιο αριθμό μεταξύ 0 και 3 συμπεριλαμβανομένων, το πλήθος των αριστερόστροφων περιστροφών του κομματιού.

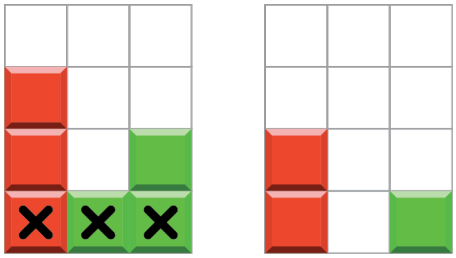
Οι συναρτήσεις `getPosition` και `getRotation` θα καλούνται μόνο μετά τη συνάρτηση `new_figure`.

Παράδειγμα

- βαθμολογητής που σας δίνεται ως υπόδειγμα κάνει τις εξής κλήσεις:
  - `init(3)` Θα υπάρχουν συνολικά τρία κομμάτια.
  - `new_figure(1)`Ένα κομμάτι τύπου 1 πέφτει από την κορυφή του "πηγαδιού".
  - `getPosition()` επιστρέφει 0. Αυτό σημαίνει ότι ο παίκτης θέλει να τοποθετήσει το κομμάτι στην αριστερότερη στήλη του "πηγαδιού".
  - `getRotation()` επιστρέφει 1 (ή 3). Αυτό σημαίνει ότι ο παίκτης θέλει να περιστρέψει το κομμάτι, ώστε να του δώσει κατακόρυφο προσανατολισμό.
  - Το κομμάτι πέφτει στον πάτο του "πηγαδιού" με το εξής αποτέλεσμα:



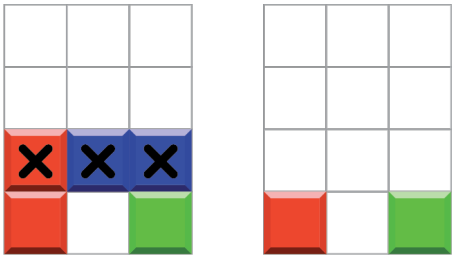
  - `new_figure(2)`. Ένα κομμάτι τύπου 2 πέφτει από την κορυφή του "πηγαδιού".
  - `getPosition()` επιστρέφει 1.
  - `getRotation()` επιστρέφει 1.
  - Μετά που θα πέσει το δεύτερο κομμάτι, η πρώτη γραμμή θα γεμίσει, οπόταν θα εξαφανιστεί με το εξής αποτέλεσμα:



  - `new_figure(1)` Ένα κομμάτι τύπου 1 πέφτει από την κορυφή του

"πηγαδιού".

- `getPosition()` επιστρέφει 1.
- `getRotation()` επιστρέφει 0 (ή 2).
- Μετά που θα πέσει το τρίτο κομμάτι, η δεύτερη γραμμή θα γεμίσει, οπόταν θα εξαφανιστεί με το εξής αποτέλεσμα:



Υποπροβλήματα

Σε όλα τα υποπροβλήματα ισχύει $n \leq 1000$.

Υποπρόβλημα 1: (7 πόντοι) Όλα τα κομμάτια είναι τύπου 1,

Υποπρόβλημα 2: (13 πόντοι) Όλα τα κομμάτια είναι τύπου 2

Υποπρόβλημα 3: (21 πόντοι) Όλα τα κομμάτια είναι τύπου 3

Υποπρόβλημα 4: (53 πόντοι) Τα κομμάτια μπορεί να είναι οποιουδήποτε τύπου.

Υπόδειγμα βαθμολογητή (Sample Grader)

- βαθμολογητής που σας δίνεται ως υπόδειγμα, διαβάζει την είσοδό του με την εξής μορφή:
  - γραμμή 1: Ένας ακέραιος n, το πλήθος των κομματιών.
  - γραμμή 2: n ακέραιοι αριθμοί: οι τύποι των κομματιών.

Σημείωση για τις γλώσσες προγραμματισμού

Χρησιμοποιήστε τα υποδείγματα αρχείων (template files) που σας δίδονται, για λεπτομέρειες υλοποίησης στην επιλεγμένη γλώσσα προγραμματισμού.