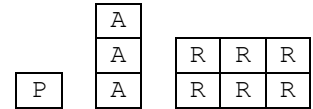


FONDAMENTI DI INFORMATICA I

FOND. DI INFORMATICA E PROGRAMMAZIONE A OGGETTI

Un *Tetris* è un tipo di dato astratto che implementa una versione molto esemplificata del popolare videogioco omonimo. Nel *Tetris* alcune forme geometriche piovono dall'alto all'interno di una scatola di dimensioni fisse (25 caselle disposte su 5 righe e 5 colonne nel nostro caso). La scatola ha un tetto, delle pareti laterali invisibili e un pavimento. Le forme, cadendo, vanno ad adagiarsi il più in basso possibile, fino a fermarsi immediatamente prima di una forma precedentemente inserita oppure sul pavimento.

In questa versione semplificata del gioco le forme *non possono essere né ruotate né traslate* durante la loro caduta e le uniche forme che possono essere inserite sono il punto (P), l'asta (A) ed il rettangolo (R), aventi forma e dimensione riportate qui a fianco.



Implementare le seguenti operazioni che possono essere effettuate su un *Tetris*:

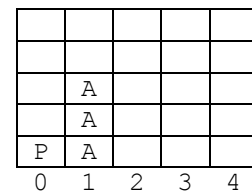
✓ **Tetris t**

Costruttore che inizializza un *Tetris* di 5 righe e 5 colonne. Inizialmente *t* non contiene forme.

✓ **t.ins(f,j)**

Funzione che aggiunge la forma *f* (dove *f* è uno dei seguenti caratteri: 'P', 'A', 'R') al *tetris t* a partire dalla colonna *j* (le colonne sono numerate a partire da 0). Se l'operazione fallisce, la funzione lascia il *Tetris* inalterato e restituisce *false*. Restituisce *true* nei casi di successo. Un esempio è il seguente:

```
Tetris t;
t.ins('P',0); // successo
t.ins('A',1); // successo
t.ins('R',3); // fallisce per mancanza di spazio a destra
```



✓ **cout << t**

Operatore di uscita per il tipo *Tetris*. Visualizza lo stato del gioco, ossia il contenuto della scatola, in un formato ben preciso che prevede di mostrare dei trattini per il tetto ed il pavimento ed i numeri da 0 a 4 sotto al pavimento ad indicare ciascuna colonna del gioco. Nel seguito sono mostrati tre esempi di *cout* di oggetti *Tetris*.

Uscita a video di *cout<<t*
prima degli inserimenti

01234

Uscita a video di *cout<<t*
dopo gli inserimenti

A

A

PA

01234

✓ **!t**

Operatore di negazione logica che restituisce il numero di righe completamente vuote presenti nell'oggetto *t* (5 e 2, rispettivamente, nei due casi precedenti).

✓ **t1==t2**

Operatore di confronto che restituisce *true* nel caso in cui i due oggetti abbiano la stessa sagoma (ossia presentino le stesse caselle vuote nelle medesime posizioni). Ad esempio i due oggetti seguenti hanno la stessa sagoma.

P

P

P

01234

A

A

A

01234

✓ **t.elimina()**

Funzione che elimina la prima riga piena ad iniziare dal basso e fa scendere le soprastanti. Questa operazione viene ripetuta finché vengono trovate righe piene. Qui sotto viene mostrato un esempio dell'esito della chiamata della funzione. Nell'esempio si noti anche che, per ragioni di semplicità, la funzione si limita a cancellare le righe piene ed a far scendere le soprastanti, evitando di far cadere oggetti che dovessero rimanere appesi nel vuoto a seguito dell'operazione, come invece avviene nel videogioco originale (infatti nell'esempio sottostante la P di posizione 2 rimane appesa nel vuoto).

cout<<t3 prima della elimina

P
RRRAA
RRRAA
P AA

01234

cout<<t3 dopo la elimina

P
P AA

01234

Utilizzando il linguaggio C++, realizzare il tipo di dato astratto `Tetris` definito dalle precedenti specifiche.

NOTE SULLO SVOLGIMENTO DELLA PROVA PRATICA

AVVIO E IDENTIFICAZIONE

- Avviare la macchina in modalità diskless, scegliere “Fondamenti di Informatica I” ed effettuare il login:
nome: studenti
password: studenti
- Aprire un terminale e spostarsi sulla cartella ‘elaborato’ (`$ cd ~/elaborato`). Si utilizzi il comando `pwd` per verificare che ci si trovi nella cartella corretta `/home/studenti/elaborato`.
- Dare il comando `$ ident`, sempre da dentro la cartella. Lo script richiede i propri dati (cognome, nome, numero di matricola e password (la password **non va dimenticata** in quanto è indispensabile per scaricare da internet il proprio elaborato a consegna avvenuta). Il comando `ident` crea il file *matricola.txt* nella cartella corrente. Lo script può essere lanciato più volte, in tal caso il file *matricola.txt* viene sovrascritto. Per verificare che il file sia stato creato e che il contenuto sia quello giusto dare il comando (la password è codificata):
`$ cat /home/studenti/elaborato/matricola.txt`
- A questo punto il docente verifica che tutti gli studenti abbiano effettuato l'identificazione, dopodiché provvede a inviare i seguenti file nella cartella `elaborato` del proprio PC: *compito.h*, *compito.cpp*, *main.cpp*. Controllare pertanto che questi file, insieme al file *matricola.txt*, siano presenti sul proprio elaboratore.

SVOLGIMENTO DELLA PROVA

- Definire ed implementare il tipo di dato astratto richiesto e le relative funzioni nei file *compito.h* e *compito.cpp*. Il file *main.cpp* contiene la funzione principale `main()` ed è utilizzato dallo studente per testare la sua implementazione della classe. Il file *main.cpp* può essere modificato a piacere. In sede di valutazione dell'elaborato verrà considerato **esclusivamente il contenuto dei file *compito.h* e *compito.cpp*** ed è pertanto **vietato cambiare nome a tali file**. Per compilare e linkare dare il comando:
`$ g++ main.cpp compito.cpp` (eseguibile invocabile tramite `$ ./a.out`)
(utilizzare `g++ -g` per includere le informazioni di debug qualora si intenda debuggare con `ddd`).

PER CONSEGNARE O RITIRARSI

Recarsi dal docente avendo preso nota dell'identificativo della macchina (g34, s23, ...).

REGOLA PER LA CORREZIONE

Per il superamento della prova pratica è **obbligatorio** aver definito ed implementato correttamente la classe e le seguenti funzioni:

- costruttore (`Tetris t`)
- inserisci forma (`t.ins(f,j)`)
- stampa a video (`cout<<t`)

Il codice di altre funzioni che impedisca la compilazione deve essere messo come commento. Tali funzioni verranno comunque valutate in sede di correzione della prova.