

**FONDAMENTI DI INFORMATICA I**  
**FOND. DI INFORMATICA E PROGRAMMAZIONE A OGGETTI**

Un `WordTetris` è una variante del famoso videogioco del Tetris, in cui a piovere dall'alto sono parole invece che forme geometriche. Le parole, che cadono in una griglia  $n \times m$ , si adagiano il più in basso possibile: alcune lettere raggiungeranno il fondo, altre sopra a lettere di parole precedentemente inserite. Inoltre, come specificato nel seguito, una lettera che si adagi sopra ad una lettera uguale provoca la cancellazione di entrambe. Si può assumere che le parole siano tutte composte da *lettere maiuscole*.

Implementare la classe `WordTetris` in modo tale da potervi effettuare le seguenti operazioni:

**PRIMA PARTE** (*qualora siano presenti errori di compilazione, collegamento o esecuzione in questa parte, l'intera prova verrà considerata insufficiente e pertanto non verrà corretta*)

✓ **`WordTetris w;`**

Costruttore che crea un `WordTetris` di 3 righe e 8 colonne. Inizialmente non sono presenti parole.

✓ **`WordTetris w(nRig, nCol);`**

Costruttore che crea un `WordTetris` di `nRig` righe e `nCol` colonne, anche in questo caso esso è inizialmente vuoto.

✓ **`w.ins(word, pos);`**

Operazione che aggiunge le singole lettere della parola specificata dalla stringa `word` nel Tetris, ove possibile, facendo in modo che esse occupino la posizione libera più in basso possibile su ciascuna colonna, a partire dalla colonna `pos` (`pos=1` indica la prima colonna). Nel caso in cui la lettera *i*-esima dovesse adagiarsi sopra ad una lettera uguale precedentemente inserita, le due lettere si cancellano a vicenda. Può capitare che una parola non possa essere inserita per intero, qualora il numero di righe o di colonne non siano sufficienti: in tal caso vengono inserite solo le lettere che cadono all'interno della griglia. La funzione restituisce `false` e lascia l'oggetto invariato solo nel caso di `pos <= 0` o maggiore del numero di colonne, negli altri casi restituisce `true`, anche qualora non generasse alcun cambiamento visibile sul tetris (ad esempio nel caso in cui l'inserimento non avvenga per insufficienza di righe).

**Esempio:**

Se dopo aver inserito in un `WordTetris 3x8` la parola "QUEL" in posizione 5 e "RAMO" in posizione 1 si inserisce la parola "DEL" in posizione 6, solo la lettera "D" verrebbe aggiunta sopra alla "U", mentre "EL" non verrebbero inserite (inoltre verrebbero cancellate "EL" della parola "QUEL" precedentemente inserita).

<pre>1  2  3 RAMOQUEL -+-----   12345678</pre>	(situazione intermedia)	<pre>1  2       DEL 3 RAMOQUEL -+-----   12345678</pre>	<b>situazione finale</b>	<pre>1  2       D 3 RAMOQU -+-----   12345678</pre>
--	----------------------------	---	------------------------------	---

✓ **`cout<<w;`**

Operatore di uscita per il tipo `WordTetris`. Un esempio di uscita è il seguente:

```
1|
2|L G D
3|R M QU
-+-----
 |12345678
```

dove viene mostrato il `WordTetris 3x8` di sopra, in cui è stata inserita anche la parola "LAGO" in posizione 1.

**SECONDA PARTE** (*si invita a mettere sotto commento le operazioni di questa seconda parte che dovessero impedire la compilazione, il collegamento o la corretta esecuzione del codice*)

✓ **`WordTetris(const WordTetris &w);`**

Costruttore di copia per la classe `WordTetris`.

✓ **w%=k**

Operatore di modulo ed assegnamento che, qualora  $k$  sia minore o al più uguale sia del numero di righe che del numero di colonne, modifica le dimensioni del `WordTetris`  $w$  trasformandolo in uno  $k \times k$ . Il contenuto del nuovo tetris coincide con il contenuto del quadrato  $k \times k$  in basso a sinistra del tetris originale.

**Esempio.** Sia  $w$  il seguente `WordTetris`:

```
1|C M C
2|L GOID
3|ROMDQUHE
-+-----
 |12345678
```

il risultato dopo  $w\%=2$  è:

```
1|L
2|RO
-+--
 |12
```

✓ **~WordTetris()** ;

Distruttore.

Mediante il Linguaggio C++, realizzare il tipo di dato astratto `WordTetris`, definito dalle precedenti specifiche. **Individuare le eventuali situazioni di errore e metterne in opera un corretto trattamento.**

## 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 al prompt spostarsi sulla cartella ‘elaborato’ (`$ cd ~/elaborato`). Si utilizzi il comando `pwd` per verificare che ci si trovi nella cartella corretta `/home/studenti/elaborato`.
- Sempre al prompt 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 abbiamo 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, ...).