

Il tipo astratto `Memory` rappresenta una versione digitale dell'omonimo gioco. Esso consiste in un tabellone di $N \times N$ caselle, ciascuna contenente tessere raffiguranti una delle seguenti immagini: gatto, cane, serpente, pavone e tigre. Le righe e le caselle del tabellone sono indicizzate a partire da 0. Il giocatore dovrà girare le tessere a due a due nell'ottica di trovare tutte le coppie di immagini uguali, collezionando il punteggio più alto possibile. Implementare le seguenti operazioni che possono essere effettuate su un `Memory`:

--- **Metodi invocati nella PRIMA PARTE di main.cpp:** ---

✓ **`Memory m(dim) ;`**

Costruttore di default che inizializza un `Memory m` avente tabellone di dimensione `dimxdim` caselle. Inizialmente non contiene tessere ed il punteggio è 0 (si veda l'operatore `<<` per come rappresentare le caselle prive di tessere).

✓ **`cout << m;`**

Operatore di uscita per il tipo `Memory`. La stampa a video ha la seguente struttura (le tessere sono mostrate girate per fini didattici ed agevolare lo studente):

```
- T G
P C P
G T C
```

Punteggio: 5

Essa rappresenta un `Memory 3x3` dove solo una casella non contiene tessere, ovvero quella in posizione (0,0). Vi sono poi due tessere raffiguranti una tigre, posizionate in (0,1) e (2,1), due raffiguranti un pavone, posizionate in (1,0) e (1,2), e due raffiguranti un gatto, posizionate in (0,2) e (2,0), e due raffiguranti un cane, locate in (1,1), e (2,2). L'attuale punteggio del giocatore è 5.

Tra ogni colonna del tabellone stampata a video vi è 1 solo spazio. Non vi sono spazi tra le righe dello stesso. Vi è una riga vuota tra la matrice e la riga del punteggio.

✓ **`m.inserisci(tipo, riga1, colonna1, riga2, colonna2);`**

Funzione membro che aggiunge al tabellone due tessere in posizioni (`riga1`, `colonna1`) e (`riga2`, `colonna2`) raffiguranti entrambe un preciso animale specificato dal `tipo` di tessera. `tipo` è un carattere maiuscolo corrispondente all'iniziale dell'animale raffigurato nella tessera. Una tessera già presente nel tabellone non può essere rimpiazzata da una nuova. Le due tessere o vengono inserite entrambe in due celle diverse del tabellone (ed entrambe libere) oppure non vengono inserite affatto. In caso di input erranei, la struttura dati deve rimanere inalterata.

✓ **`m.riassumi();`**

Riassunto dell'attuale stato del `Memory m` attraverso la stampa a video del numero di coppie presenti per ogni tipo (per l'implementazione può essere utile aver prima letto il testo della seconda parte). Segue un esempio.

```
Gatto: 3
Cane: 0
Serpente: 1
Pavone: 1
Tigre: 0
```

L'ordine in cui le tipologie di tessere sono riportate deve essere esattamente lo stesso di cui sopra. Tra i due punti ed il numero di coppie vi è 1 spazio. In caso non vi siano coppie rimaste, la funzione deve semplicemente stampare a video la stringa VITTORIA! seguita da una nuova linea.

✓ `m.flip(riga1, colonna1, riga2, colonna2);`

Funzione membro che gira due distinte tessere del Memory `m`, rispettivamente in posizioni `(riga1, colonna1)` e `(riga2, colonna2)`. Se le due tessere rappresentano lo stesso animale, il giocatore guadagna un punto, le due tessere vengono rimosse e la funzione ritorna `true`. In tutti gli altri casi, la funzione restituisce `false`. Qualora le due tessere esistano ma siano diverse, il giocatore perde anche un punto.

✓ `~m;`

Distruttore.

--- Metodi invocati nella SECONDA PARTE di main.cpp: ---

✓ `Memory m1(m);`

Costruttore di copia che istanzia il Memory `m1` con un tabellone identico a quello del Memory `m` ma con punteggio nullo.

✓ `m+m1;`

Operatore di somma tra Memory che ne genera uno nuovo Memory con punteggio 0 e avente tabellone identico a quello di `m` ma le cui celle vuote sono riempite con copie delle corrispondenti tessere di `m1`. In caso i due Memory abbiano dimensioni diverse, assumere che `m` ed `m1` siano allineati rispetto all'angolo in alto a destra. Segue un'esemplificazione della somma tra un Memory 3x3 ed uno 2x2.

```
- T S   - T -   C S
P - P = P - P + - -
G T C   G T C
```

Notare che la somma in questo caso non gode della proprietà commutativa:

```
C S   C S   - T -
- P = - - + P - P
          G T C
```

N.B. A seguito della somma, il Memory risultante può presentare tessere **NON** accoppiate, come peraltro evidenziato da entrambi gli esempi sopra mostrati.

✓ `m>>i;`

Operatore di shift destro che ruota il Memory `m` in senso orario di un angolo pari a $i \cdot \pi/2$. L'operatore restituisce un riferimento ad `m` dopo l'avvenuta modifica. N.B.: `i` può essere negativo.

Mediante il linguaggio C++, realizzare il tipo di dato astratto **Memory**, definito dalle precedenti specifiche. Non è permesso utilizzare funzionalità della libreria STL come il tipo `string`, il tipo `vector`, il tipo `list`, ecc. **Gestire le eventuali situazioni di errore.**

USCITA CHE DEVE PRODURRE IL PROGRAMMA

--- PRIMA PARTE ---

Test del costruttore

- - -
- - -
- - -

Punteggio: 0

Test della inserisci

- - -
- - -
S - S

Punteggio: 0

Test della riassumi

Gatto: 0
Cane: 0
Serpente: 1
Pavone: 0
Tigre: 0

Test della flip: 1

Altro test della flip: 0

- - -
- - -
- - -

Punteggio: 1

Altro test della riassumi

VITTORIA!

Inserimenti leciti

Altro test della flip: 0

P T C
P - C
S T S

Punteggio: 1

Test del distruttore

--- SECONDA PARTE ---

Test del costruttore di

copia

P T C

P - C

S T S

Punteggio: 0

Altro test del distruttore

Distruzione avvenuta

Costruzione di un Memory 2x2

Altro test della flip: 0

S S

- -

Punteggio: 0

Test operatore +

S S

- C

Punteggio: 0

P T C

P - C

S T S

Punteggio: 0

Test operatore >> con i=1

S P P

T - T

S C C

Punteggio: 1

Test operatore >> con i=-2

- -

S S

Punteggio: 0

Note per la consegna:

Affinché l'elaborato venga considerato valido, il programma **deve** produrre almeno la prima parte dell'output atteso. In questo caso, i docenti procederanno alla valutazione dell'elaborato **solo se** lo studente avrà completato l'autocorrezione del proprio elaborato.

In **tutti** gli altri casi (per esempio, il programma non compila, non collega, non esegue o la prima parte dell'output non coincide con quella attesa), l'elaborato è considerato **insufficiente** e, pertanto, **non verrà corretto**.