CORSO DI LAUREA IN INGEGNERIA INFORMATICA

Un Portafoto rappresenta un portafoto da parete come quelli mostrati in figura. Esso è costituito da un numero prefissato di *colonne* lungo le quali può essere attaccato un numero arbitrario di foto. Ogni foto può essere associata (per qualsivoglia ragione) ad un'altra foto nel Portafoto. Si noti che l'associazione tra foto non è biunivoca: se una foto f è associata alla foto g, non è necessario il viceversa; g potrebbe risultare associata ad una terza foto h oppure non essere associata affatto. Implementare le seguenti operazioni che possono essere effettuate su un Portafoto:



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

✓ Portafoto p(d);

Costruttore che inizializza un Portafoto di dimensione d, ovvero avente d colonne utilizzabili. Inizialmente il Portafoto è vuoto, ovvero non contiene alcuna foto. Preoccuparsi che d sia positivo. In caso contrario, procedere per sostituzione usando la dimensione di default 3.

✓ cout << p;</pre>

Operatore di uscita per il tipo Portafoto. La stampa a video avviene come nell'esempio di seguito:



Qui è rappresentato un Portafoto a tre colonne, la prima contenente due foto, la seconda tre e la terza una. Inoltre vi sono tre associazioni di foto all'interno del Portafoto: i) la foto in posizione 1 lungo la colonna 1 è associata alla foto in posizione 1 colonna 3; ii) la foto in posizione 2 colonna 2 è associata alla foto in posizione 2 colonna 1; iii) la foto in posizione 3 colonna 2 è associata alla foto in posizione 1 colonna 1. Non vi sono altre associazioni tra foto in tutto il Portafoto. Dall'esempio si evince che la rappresentazione di una foto nelle associazioni ha la forma (c,p) dove c indicare il numero di colonna e p la posizione della foto all'interno di essa.

Il numero di "_" (underscore) che formano il supporto orizzontale del Portafoto è pari al doppio delle colonne più uno, ogni colonna è distanziata da un solo spazio, ogni foto è rappresentata con un "*" (asterisco) e lungo ogni colonna le foto sono separate da due "|" (linee verticali).

Le associazioni devono essere mostrate colonna per colonna; lungo ognuna di esse le associazioni devono essere mostrate per posizione crescente. Questo significa che prima viene mostrata (se presente) l'associazione della foto in posizione 1 della colonna 1, poi quella della foto in posizione 2 della colonna 1, e così via fino all'esaurimento della colonna stessa. A questo punto la funzione passa a riportare le associazioni della colonna successiva, ovvero la 2, ripetendo il procedimento fino a che tutte le colonne sono state visitate e le associazioni delle relative foto mostrate. Non vi sono spazi tra i caratteri.

✓ p.aggiungi(col, pos);

Funzione membro che aggiunge al Portafoto puna foto lungo la colonna col alla posizione pos.

Si noti che:

- Le foto sono aggiunte consecutivamente lungo una colonna, ovvero una foto può essere collocata in posizione 2 solo se la posizione 1 della medesima colonna è occupata.
- Qualora la posizione della nuova foto sia occupata, l'occupante e tutte le successive scorrono di una posizione verso il basso così da liberarla. L'associazione tra le foto deve essere aggiornata coerentemente.
- Qualora la posizione della nuova foto non sia ammissibile la funzione fallisce.

La funzione restituisce false in caso di fallimento, true altrimenti.

```
✓ p.associa(col, pos, col_ass, pos_ass);
```

Funzione che aggiorna la foto in posizione pos lungo la colonna col associandole la foto in posizione pos_ass lungo la colonna col_ass. Se la foto era già associata ad un'altra, questa informazione viene sovrascritta e dunque persa. Se una delle due foto non è presente nel Portafoto p, la funzione restituisce false lasciando inalterata la struttura dati, altrimenti true.

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

✓ p.elimina(col, pos);

Funzione membro che elimina una foto in posizione pos lungo la colonna col dal Portafoto p e compatta la colonna facendo scorrere le eventuali foto successive verso l'alto di una posizione. Se una più foto sono associate a quella da eliminare, le prime devono essere modificate in modo da risultare non più associate ad alcuna foto. La funzione restituisce true in caso di corretta eliminazione, altrimenti false.

√ p=q;

Operatore di assegnamento per Portafoto.

✓ ~Portafoto();

Distruttore.

Mediante il linguaggio C++, realizzare il tipo di dato astratto Portafoto, 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**.

SUGGERIMENTO 1: Può tornare molto comodo implementare una funzione di utilità get_foto (col, pos) che restituisca un riferimento alla foto in posizione pos lungo la colonna col.

SUGGERIMENTO 2: Ragionare bene su come rappresentare l'associazione tra foto all'interno della struttura dati. Una buona scelta può facilitare molto l'implementazione, specialmente considerando le funzioni aggiungi ed elimina.

USCITA CHE DEVE PRODURRE IL PROGRAMMA

--- PRIMA PARTE ---Test del costruttore Associazioni: Test della aggiungi 1 0 I I II I II IAssociazioni: Test della associa 0 I I II I I* * * \perp

Associazioni:

 $(1,1) \rightarrow (3,1)$

 $(2,2) \rightarrow (1,2)$

 $(2,3) \rightarrow (1,1)$

Test dell'aggiornamento delle associazioni dopo aggiunta



Associazioni:

 $(1,1) \rightarrow (3,2)$

 $(2,2) \rightarrow (1,2)$

 $(2,3) \rightarrow (1,1)$

```
--- SECONDA PARTE ---
Test della elimina
0
 I I I
 I I I
   1
Associazioni:
(1,1) \rightarrow (3,2)
(2,2) \rightarrow (1,2)
(2,3) \rightarrow (1,1)
Test dell'aggiornamento delle associazioni dopo eliminazioni
 I I I
 | | |
   1 1
Associazioni:
(1,1) \rightarrow (3,2)
(2,2) \rightarrow (1,1)
Operatore di assegnamento
 1
 1
        1 1
        1
Associazioni:
```

 $(4,1) \rightarrow (1,1)$ $(4,2) \rightarrow (5,2)$ $(5,3) \rightarrow (5,2)$

Test del distruttore Oggetto distrutto

Nota finale. 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 (sia della prima che della seconda parte).

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