FONDAMENTI DI INFORMATICA **26 FEBBRAIO 2020**

CORSO DI LAUREA IN INGEGNERIA BIOMEDICA **FILA A**

Un `Museo` è suddiviso in 2 sale. In ogni sala è possibile esporre fino a N opere. Ogni opera è caratterizzata da una descrizione, che consiste in una stringa minore o uguale a 100 caratteri. Un'opera può essere aggiunta ad una sala del museo, rimossa da una sala oppure spostata da una sala ad un'altra. La numerazione delle sale parte da 0. Implementare il tipo `Sala` come una lista di `Opere`. Implementare le seguenti operazioni che possono essere compiute su un `Museo`:

✓ **`inizializzaMuseo(M, N);`** **[4pt]**

Funzione che inizializza un museo `M`, tale che ogni sala può mantenere al più N opere. Se N è un numero minore o uguale a 0, ogni sala può esporre al più 10 opere. Inizialmente, tutte le sale sono vuote.

✓ **`aggiungiOpera(M, str);`** **[5pt]**

Funzione booleana che aggiunge l'opera con descrizione `str` alla sala del museo `M` *con meno opere*. Nel caso in cui le sale abbiano lo stesso numero di opere, la scelta della sala è arbitraria. Se tutte le sale sono piene oppure se esiste già un'opera con descrizione `str` il museo è lasciato inalterato. Se l'operazione ha successo (ovvero, se è possibile inserire l'opera) la funzione restituisce `true`, altrimenti la funzione restituisce `false`.

✓ **`stampaMuseo(M);`** **[3pt]**

Funzione che stampa a video tutte le descrizioni delle opere nelle varie sale del museo `M`. Le opere vengono stampate per sala, su righe separate, e *in ordine di inserimento dalla meno recente alla più recente*. Se una sala non ha opere, stampa il numero della sala seguito dalla stringa `"---"`. Nel seguente esempio, `DipintoA` è l'opera inserita per prima nella `Sala 0`:

```
Sala 0: DipintoA, SculturaA
Sala 1: DipintoB
```

✓ **`cercaOpera(M, op);`** **[3pt]**

Funzione che verifica se esiste all'interno del museo `M` l'opera con descrizione `op`. Nel caso esista un tale opera, la funzione restituisce il numero della sala in cui si trova, altrimenti restituisce `-1`.

✓ **`spostaOpere(M, op1, op2);`** **[5pt]**

Funzione booleana che sposta l'opera con descrizione `op1` al posto dell'opera con descrizione `op2` e viceversa all'interno del museo `M`. Se l'opera con descrizione `op1` e/o con descrizione `op2` non esiste, il museo è lasciato inalterato e la funzione restituisce `false`, altrimenti restituisce `true`.

✓ **`rimuoviOpera(M, op);`** **[5pt]**

Funzione booleana che rimuove l'opera `op` dal museo `M`. Se l'eliminazione ha successo la funzione restituisce `true`, `false` altrimenti.

Mediante il linguaggio C++, implementare il tipo `Museo` definito dalle precedenti specifiche utilizzando le **strutture**. Individuare eventuali situazioni di errore, e metterne in opera un corretto trattamento.

Continua sul retro…

Esempio di funzione main()

```
int main(){

     Museo M;

     inizializzaMuseo(M, 5);
     stampaMuseo(M);                    // Sala 0: ---
                                        // Sala 1: ---

     aggiungiOpera(M, "DipintoA");
     aggiungiOpera(M, "DipintoB");
     aggiungiOpera(M, "SculturaA");
     aggiungiOpera(M, "DipintoA");
     stampaMuseo(M);                    // Sala 0: DipintoA, SculturaA
                                        // Sala 1: DipintoB

     cout << cercaOpera(M, "DipintoA") << endl;        // 0
     cout << cercaOpera(M, "SculturaC") << endl;       // -1


     spostaOpere(M, "SculturaA", "DipintoB");
     stampaMuseo(M);                    // Sala 0: DipintoA, DipintoB
                                        // Sala 1: SculturaA

     rimuoviOpera(M, "DipintoA");
     stampaMusemo(M);                   // Sala 0: DipintoB
                                        // Sala 1: SculturaA

     return0;
}
```

Domande

1. **[2pt]** Dato il numero 121 in base 3, trovare la sua rappresentazione in base 5.

2. **[2pt]** Scrivere una funzione che prende in ingresso un array di interi di dimensione N e restituisce una nuova matrice NxN in cui tutti i valori sono inizializzati a zero tranne sulla diagonale secondaria, che è inizializzata con gli elementi dell'array.

3. **[1pt]** Eseguire il seguente codice e indicare il risultato delle operazioni di uscita.

```
int n = 20;
double d = 5.4;
cout << n + d << endl;
n = n + d;
cout << n << endl;
```