

ESERCIZIO 1

- VETTORI, LISTE, PILE E CODE DI DATI GENERICI -

F. MOGAVERO

Implementare quattro librerie in **Linguaggio C++** per la gestione di strutture dati dinamiche di tipo **vettore**, **lista**, **pila** e **coda** contenenti **dati generici**, ovvero interi, stringhe, ecc. In particolare, per le ultime due strutture dovranno essere sviluppate due diverse implementazioni concrete basate su (a) **vettori** con efficiente **ridimensionamento automatico** dello spazio allocato e (b) **liste**.

Le funzionalità da realizzare sono di seguito elencate:

- (1) **costruzione** e **distruzione** di una struttura dati;
- (2) operazioni di **assegnamento** e **confronto** tra istanze diverse della specifica struttura dati;
- (3) operazioni comuni ai quattro tipi di strutture dati: test di **vuotezza**; lettura della **dimensione**; **svuotamento** della struttura;
- (4) operazioni comuni a vettori e liste: **accesso non distruttivo** all'elemento iniziale, finale o avente uno specifico indice; **controllo di esistenza** di un dato valore; operazioni di **applicazione di una funzione** a tutti gli elementi (**funzioni map**); operazioni di **attraversamento** e **accumulazione di un valore** (funzioni **traverse** e **fold**);
- (5) operazioni comuni a pile e code: **inserimento** di un dato elemento; **rimozione**, **rimozione con lettura** e **lettura non distruttiva** dell'elemento immediatamente accessibile;
- (6) funzioni specifiche dei vettori: **ridimensionamento** del vettore;
- (7) funzioni specifiche delle liste: **inserimento** di un dato valore in testa o coda; **rimozione** e **rimozione con lettura** dell'elemento in testa; operazioni di gestione di un **dizionario**.

Al fine di poter testare adeguatamente il funzionamento delle librerie sopra descritte, si richiede di definire (esternamente alle stesse, in un opportuno file di test richiamato dal “main”) un insieme di procedure di **test unitario**. In aggiunta a questo, è necessario prevedere l'accesso alla funzionalità di test prevista dal docente.

Il codice sorgente prodotto dovrà seguire pedissequamente (sia nei nomi delle funzioni di libreria, sia nella strutturazione, gerarchia di classi e nei nomi delle diverse directory e file “.cpp” e “.hpp”) la forma riportata nel template Exercise1.zip associato al presente esercizio.

Per lo sviluppo delle librerie, in aggiunta alle spiegazioni del docente e al diagramma delle classi reperibile sul sito docenti d'ateneo, si faccia riferimento alla sezione 13.6 di [1] e alle sezioni 4.1, 4.2, 4.3 e 4.4 di [2].

RIFERIMENTI BIBLIOGRAFICI

- [1] Bjarne Stroustrup. *The C++ Programming Language*, Addison-Wesley, 4th Edition, 2013.
- [2] Clifford A. Shaffer. *Data Structures and Algorithm Analysis*, Edition 3.2 (C++ Version), 2013.