ESERCIZIO 1
- VETTORI E LISTE DI DATI GENERICI -

F. MOGAVERO

Implementare due librerie di funzioni in **Linguaggio C++** per la gestione di strutture dati dinamiche di tipo **vettore** e **lista** per **dati generici**, ovvero interi, float, stringhe, struct, ecc.

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 struttura dati;
(3) operazioni comuni ai due tipi di strutture dati (**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 **accumulazione di un valore** estratto dagli elementi: **funzioni fold**; test di **vuotezza**; lettura della **dimensione**; **svuotamento** della struttura);
(4) funzioni specifiche su vettore (**ridimensionamento** del vettore);
(5) funzioni specifiche su lista (**inserimento** di un dato valore in testa o coda; **rimozione** e **rimozione con lettura** dell'elemento in testa).

Al fine di poter testare adeguatamente le librerie sopra descritte, si richiede di definire (esternamente alle stesse, in un opportuno file di test richiamato dal "main") un insieme di procedure che implementi le seguenti funzionalità:

(1) **scelta della struttura** (vettore vs lista) e del relativo **tipo di dati** da gestire (interi, ecc.);
(2) **popolamento della struttura** precedentemente scelta con n valori del tipo opportuno generati casualmente, dove n è un parametro dato dall'utente in ingresso;
(3) **visualizzazione dell'elemento** iniziale, finale o avente uno specifico indice;
(4) **visualizzazione di tutti gli elementi** (effettuata per mezzo della **funzione map**);
(5) **controllo di esistenza** di un dato valore;
(6) **calcolo di una delle seguenti funzioni** (effettuato per mezzo delle **funzioni fold**) e relativa visualizzazione del risultato: somma per gli interi minori di n, prodotto per i float maggiori di n, e concatenazione per le stringhe con lunghezza minore o uguale a n, dove n è un parametro dato dall'utente in ingresso;
(7) **applicazione di una delle seguenti funzioni** a tutti gli elementi: $2n$ per gli interi, n^2 per i float, e "uppercase" per le stringhe.

Da un opportuno menu, dovrà essere inoltre possibile operare sulla struttura scelta attraverso le funzioni di libreria di cui ai punti (4) e (5). Infine, 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.