Corso Programmazione ad Oggetti
Corso di Laurea in Informatica
Esercitazione di Laboratorio del 16/10/2017

ESERCIZIO 1 Realizzare un main che letti da input alcuni numeri li inserisca in un vettore STL (http://www.cplusplus.com/reference/vector/vector/). Eseguire quindi le seguenti operazioni:

1. Stampare il contenuto del vettore scandendo il vettore con un for su un indice.
2. Stampare il contenuto del vettore scandendo il vettore usando un iteratore.
3. Stampare la dimensione del vettore.
4. Stampare la capacità del vettore.
5. Stampare il primo intero nel vettore.
6. Stampare l'ultimo intero nel vettore.
7. Stampare il penultimo intero nel vettore.
8. Ridimensionare il vettore in modo che contenga meno elementi di quelli attualmente presenti (ad esempio, se il vettore contenesse 10 elementi ridimensionare a 5, usando il metodo ***resize***). Stampare nuovamente il vettore, la sua size e la sua capacità e verificare il comportamento del metodo.
9. Ridimensionare il vettore in modo che contenga più elementi di quelli attualmente presenti (ad esempio, se il vettore contenesse 5 elementi ridimensionare a 10, usando il metodo ***resize***). Stampare nuovamente il vettore, la sua size e la sua capacità e verificare il comportamento del metodo.

ESERCIZIO 2 Realizzare un main che letti da input alcuni numeri li inserisca in una lista STL (http://www.cplusplus.com/reference/list/list/). Eseguire quindi le seguenti operazioni:

1. Stampare il contenuto della lista.
2. Stampare la dimensione della lista.
3. Stampare il primo intero nella lista.
4. Stampare l'ultimo intero nella lista.
5. Stampare il penultimo elemento presente nella lista .
6. Rimuovere gli interi duplicati e consecutivi nella lista (usare il metodo ***unique***). Stampare nuovamente il contenuto della lista e verificare i duplicati consecutivi siano stati eliminati.
7. Ordinare la lista in modo che gli interi contenuti in essa siano ordinati in ordine crescente (usare il metodo ***sort***). Stampare nuovamente il contenuto della lista e verificare che l'ordinamento sia rispettato.
8. Rimuovere tutti gli interi duplicati nella lista (usando nuovamente ***unique***). Stampare nuovamente il contenuto della lista e verificare tutti duplicati siano stati eliminati.

Ad esempio se la lista contenesse: 1 - 2 - 2 - 7 - 8 - 1 - 1 - 3 al punto 6 si otterrebbe: 1 - 2 - 7 - 8 - 1 - 3 mentre al punto 7: 1 - 1 - 2 - 3 -7 - 8 ed al punto 8: 1 - 2 - 3 - 7 - 8.

ESERCIZIO 3

Utilizzare la classe ***Contatto***, allegata alla traccia, per realizzare una classe ***GestoreContatti***.

Dotare la classe di un dato private che sia una lista di contatti, e dei seguenti metodi che permettano di gestire la lista:

- **void stampaContatti ()**, che stampi su standard output il contenuto della lista.
- **bool inserisciContatto (const Contatto& c)**, che aggiunga il contatto c, alla fine della lista, se non esiste già un contatto uguale. Restituire true se è stato inserito, false altrimenti.
- **int numeroContatti ()**, che restituisca il numero di contatti presenti nella lista.

Corso Programmazione ad Oggetti
Corso di Laurea in Informatica
Esercitazione di Laboratorio del 16/10/2017

- **void ordinaContatti ()**, che ordini la lista di contatti.
- **bool eliminaContattoConRemove (const Contatto& c)**, che elimini dalla lista il contatto c, se presente. Utilizzare a questo scopo il metodo *remove* di list (http://www.cplusplus.com/reference/list/list/remove/). Restituire true se è stato eliminato, false altrimenti.
- **bool eliminaContattoConErase (const Contatto& c)**, che come il metodo precedente, elimini dalla lista il contatto c, se presente. Tuttavia, in questo caso, si utilizzi il metodo *erase* di list. Si faccia attenzione al fatto che il metodo *erase* invalida gli iteratori che fanno riferimento agli elementi rimossi (http://www.cplusplus.com/reference/list/list/erase/). Restituire true se è stato eliminato, false altrimenti.
- **vector<string> trovaNumeri (const string& nome, const string& cognome)**, che restituisca un vettore contenente i numeri di telefono memorizzati nella lista in corrispondenza dei contatti con il nome ed il cognome ricevuti come parametri.
- **string trovaCognomePiuFrequente ()**, che restituisca una stringa corrispondente al cognome con il maggior numero di occorrenze nella lista. Se la lista è vuota restituire una stringa vuota.
- **bool verificaDueContattiStessoTelefono ()**, che restituisca true se nella lista sono presenti almeno due contatti con nome e/o cognome diversi, ma stesso numero di telefono, false altrimenti.

Realizzare un main in cui viene creato un oggetto di tipo ***GestoreContatti***, e viene controllata la correttezza dei metodi implementati. Verificare, inoltre, che l'effetto dei due metodi per l'eliminazione sia lo stesso.

Alcuni suggerimenti per lo svolgimento:

- Non utilizzare caratteri accentati per nomi di campi, variabili o metodi.
- Per il carattere tilde (~) occorre digitare: ALT + 126 su sistemi Windows, ALT gr + ì su sistemi Linux, ALT + 5 su sistemi Mac OS X.
- Per ogni esercizio realizzare una cartella contenente tutti i file creati.
- Si consiglia di utilizzare un semplice editor di testo (ad esempio, gedit), e compilare da linea di comando. Per compilare occorre passare tutti i file sorgente (.cpp) al compilatore.
- Minimizzare le inclusioni.