

Progetto – "Archivio città e mappa"

Specifiche funzionali

Realizzare un programma C++ che offra funzioni di base per la gestione di un archivio di città e della relativa mappa dei collegamenti. In particolare, il programma dovrà fornire le seguenti funzioni:

Gestione archivio delle città

(a) **immissione** nuova città; per ogni città vengono fornite (almeno) le seguenti informazioni:

- nome della città (anche nomi composti; es. Reggio Emilia)
- provincia (sigla; es. PR)
- numero abitanti (facoltativo)¹
- elenco principali attrazioni turistiche nella città (max. 10 stringhe).

(b) **modifica** città (individuata tramite sottostringa del nome). È possibile modificare soltanto le informazioni relative al numero abitanti e all'elenco delle attrazioni (aggiunta e cancellazione).

(c) **cancellazione** città (individuata tramite sottostringa del nome); N.B. la cancellazione è possibile soltanto se la città non compare (più) nella mappa.

(d) **ricerche**

- visualizzazione dei dati di una data città (individuata tramite sottostringa del nome)
- visualizzazione del nome di tutte le città appartenenti ad una certa provincia
- visualizzazione del nome di tutte le città con meno di n abitanti
- visualizzazione del nome di tutte le città che contengono una certa attrazione turistica (individuata tramite sottostringa).

Gestione mappa

(a) **aggiunta** alla mappa di una nuova città e degli eventuali collegamenti con altre città presenti nella mappa. N.B. le città sulla mappa sono sempre individuate tramite sottostringa del nome e devono essere già presenti nell'archivio delle città.

(b) **aggiunta** alla mappa di un nuovo **collegamento**; per ogni collegamento vengono fornite (almeno) le seguenti informazioni:

- nome delle due città agli estremi del collegamento
- lunghezza del collegamento (in km)
- tipo del collegamento (autostrada, strada normale)
- se a pagamento, costo medio (in Euro) (facoltativo)
- tempo medio di percorrenza (in ore e minuti) (facoltativo).

(c) **inserimento** di una città su un collegamento già esistente: vengono fornite le due città limitrofe alla città da inserire e le distanze tra quest'ultima e le due limitrofe. N.B. se il collegamento tra le due città specificate non è diretto (e cioè contiene almeno un'altra città) il programma chiede di ripetere l'input dei dati.

(d) **ricerche**

- individuazione del **cammino più breve** tra due città presenti sulla mappa; viene visualizzato il percorso, la lunghezza, il costo e il tempo di percorrenza totali (gli ultimi due se disponibili)
- come ricerca (i), ma utilizzando soltanto strade normali
- visualizzazione del nome di tutte le città che distano da una data città non più di k chilometri.

(e) **cancellazione** di un collegamento o di una città dalla mappa; la cancellazione di un collegamento tra due città è possibile solo se non coinvolge altre città; la cancellazione di una città è possibile soltanto quando non ci sono più collegamenti su quella città.

L'interazione con l'utente avviene tramite **menù a più livelli**. Il programma provvede anche ad effettuare controlli sulla **correttezza dei dati** immessi (ad esempio, codice dell'operazione scelta, validità lunghezza, ecc.).

¹ Facoltativo significa che l'utente può non fornire alcuna informazione per questo attributo; l'attributo è comunque presente, anche se senza un valore significativo.

Note di progettazione

- Le entità *città*, *collegamento*, *mappa*, *archivio_città* e *tempo* devono essere definite come altrettante **classi**. Valutare se definire *strada_normale* e *autostrada* come classi derivate da una classe base *collegamento*.
- Tutti i dati relativi alla *mappa* e all'*archivio_città* devono essere salvati su altrettanti **file** su richiesta esplicita dell'utente e/o automaticamente al termine del programma, e caricati da file in memoria principale all'inizio dell'esecuzione del programma.

Note di implementazione

- Si richiede di realizzare la gestione di eventuali situazioni anomale tramite i meccanismi di **gestione eccezioni** del C++ ove opportuno.
- Le diverse operazioni (aggiunte, ricerche, modifiche, ...) devono essere svolte su dati memorizzati in **memoria principale** (ad esempio, array) e non direttamente su file.
- E' vietato: usare dichiarazioni `friend`; usare `goto`; usare variabili globali (non motivate); definire classi con solo parte `public`; usare ereditarietà a sproposito; usare librerie grafiche, usare sistemi di gestione basi di dati.