

Fondamenti di Informatica T1

Prova Pratica - Settembre 2019

Prima di cominciare: si scarichi il file **StartKit.zip** contenente i file di esempio.
Avvertenze per la consegna: nominare i file sorgenti come richiesto nel testo del compito, apporre all'inizio di ogni file sorgente un commento contenente i propri dati (**cognome, nome, numero di matricola**) e il **numero** della prova d'esame. Al termine, **consegnare tutti i file sorgente ed i file contenuti nello StartKit**.
Rispettare le specifiche, in particolare inserire le funzioni nei file specificati fra parentesi dopo il nome della funzione. Chi non rispetta le specifiche sarà opportunamente penalizzato. **NON SARANNO CORRETTI** gli elaborati che presenteranno un numero "non ragionevole" di errori di compilazione.
Consiglio: per verificare l'assenza di *warnings*, effettuare di tanto in tanto un *Rebuild All*.

La compagnia di trasporti ferroviari TPIU ha deciso di realizzare un sistema informatizzato per la gestione dei collegamenti via treno offerti e delle prenotazioni effettuate. Il sistema memorizza le tratte offerte dalla compagnia in un opportuno file di testo *collegamenti.txt* (presente nello start kit). Ogni riga di tale file corrisponde ad una tratta che collega due città (senza scali) ed è composta dai seguenti campi:

codiceTreno cittàPartenza cittàDestinazione durata prezzo

dove *codiceTreno* è una stringa di caratteri alfanumerici di 5 caratteri, *cittàPartenza* è una stringa di 3 caratteri che identifica ovviamente la città da dove parte il treno mentre *cittàDestinazione* è una altra stringa di 3 caratteri che specifica la destinazione; la *durata*, un intero, è espressa in minuti e infine il *prezzo* è un float che indica il costo di un biglietto singolo per il singolo viaggio (quindi sola andata). Si noti inoltre che se nel file è presente una tratta che collega le città A e B, allora è automaticamente prevista la possibilità di effettuare il tragitto dalla città B ad A, senza che sia necessaria la presenza di una riga nel file che ne preveda esplicitamente l'esistenza.

Non è noto a priori quanti collegamenti siano presenti nel file.

In un altro file di testo chiamato *prenotazioni.txt* (presente nello start kit) sono invece salvate le informazioni relative alle prenotazioni dei treni effettuate. Le righe di questo file sono formate nel seguente modo:

**giornoAcq meseAcq annoAcq giornoViag meseViag annoViag
codiceTreno cliente AR quantita totale**

dove i primi sei campi rappresentano la data della prenotazione (cioè dell'acquisto dei biglietti) e quella del viaggio da effettuare, *codiceTreno* specifica la tratta desiderata, *cliente* è una stringa senza spazi contenente il nome di chi ha effettuato la prenotazione (al massimo 20 caratteri), il campo *AR* è un carattere che specifica se la prenotazione riguarda la sola andata ('A') oppure andata e ritorno ('R'), *quantita* è un intero che indica quanti biglietti sono stati prenotati e *totale* è un float con il costo totale pagato dal cliente per la sua prenotazione. Anche in questo caso non è noto a priori il numero di prenotazioni presenti nel file.

Fondamenti di Informatica T1

Prova Pratica - Settembre 2019

Esercizio 1 - Lettura dei dati (*gestioneTratte.h/gestioneTratte.c*)

Si definisca un'opportuna struttura dati **Tratta**, al fine di rappresentare i dati registrati nel file *collegamenti.txt* come specificato sopra (per la rappresentazione delle date di acquisto e di viaggio si suggerisce di creare due **struct** *dataAcquisto* e *dataViaggio*, ciascuna contenente tre campi interi **giorno**, **mese**, **anno**), e si realizzi una procedura:

Tratta* leggiCollegamenti (char fileName[], int *dim);

che, ricevuto in ingresso il nome di un file contenente i dati dei collegamenti, legga i dati relativi e li memorizzi in un array allocato dinamicamente; la funzione deve restituire un riferimento all'area di memoria allocata (array) e la sua dimensione logica (numero di collegamenti letti) in **dim**. Come già detto, il numero di collegamenti presenti nel file non è noto a priori occorre determinarlo per allocare la memoria necessaria.

Si realizzi in seguito la procedura

void stampaTratta (Tratta t);

che, ricevuto in ingresso una **Tratta** ne stampi a video il contenuto, ovvero tutti i campi della struttura.

In maniera simile si realizzi una procedura per la lettura delle prenotazioni presenti nel file *prenotazioni.txt*, dopo aver definito una struttura **Prenotazione** adatta per contenere le relative informazioni:

Prenotazione* leggiPrenotazioni(char fileName[], int *dim);

che, ricevuto in ingresso il nome di un file contenente i dati delle prenotazioni, legga i dati relativi e restituisca un array di **Prenotazione**; questa funzione deve restituire il numero di prenotazioni lette tramite il parametro **dim**.

Nuovamente, si realizzi la procedura

void stampaPrenotazione(Prenotazione p);

che, ricevuto in ingresso una **Prenotazione** ne stampi a video il contenuto.

Si realizzino nel **main** le opportune istruzioni per verificare il corretto funzionamento delle funzioni implementate, avendo cura di non cancellarle dopo aver verificato il test ma limitandosi a commentarle (se necessario).

Fondamenti di Informatica T1

Prova Pratica - Settembre 2019

Esercizio 2 – Effettua prenotazione (*gestioneTratte.h/gestioneTratte.c*)

Si realizzi una funzione

int nuovaPrenotazione(FILE* fp, Tratta* t, int dimT);

che preso in ingresso un puntatore ad un file di testo già aperto, un array di **Prenotazione** p e un array di **Tratta** t, con le relative dimensioni (rispettivamente dimP e dimT) provveda a scrivere nel file passato come argomento la nuova prenotazione inserita dall'utente.

Tramite tastiera, l'utente dovrà inserire le date (giorno, mese, anno) della prenotazione e del viaggio, il nome del cliente, il codice della città di partenza e quello della città di arrivo, specificare se intende acquistare un biglietto di sola andata oppure andata e ritorno ed infine indicare il numero di biglietti da acquistare. La funzione dovrà controllare che esista effettivamente una tratta che collega le due città richieste, cercando nell'array **T**, in caso positivo dovrà poi calcolare il prezzo totale della prenotazione, tenendo conto del numero di biglietti richiesti e del fatto che il treno sia di sola andata oppure andata e ritorno – in quest'ultimo caso, se la tratta richiesta è disponibile, il prezzo del biglietto per l'andata e il ritorno è pari al prezzo del biglietto singolo moltiplicato per 1.7. Se la data della prenotazione è precedente a quella del viaggio di più di sei mesi viene applicato al prezzo totale uno sconto del 20%, se è precedente di più di 12 mesi lo sconto è del 50%. **Per semplicità, si consideri la durata di ogni mese pari a 30 giorni, e quella di un anno pari a 360 giorni.** La funzione deve ritornare il valore '0' se l'inserimento è andato a buon fine, '-1' in caso contrario.

Il candidato scriva le opportune istruzioni nel main per testare questa funzione, passandole come argomento il file *prenotazioni.txt* e controllando che venga correttamente aggiornato con la nuova prenotazione.

Esercizio 3 – Trova collegamenti (*gestioneTratte.h/gestioneTratte.c*)

Il candidato realizzi una funzione

Tratta* trovaCollegamento (char* citta, Tratta* t, int dimT, int* dimResult);

che prenda in ingresso il codice di una città, l'array contenente tutte le tratte offerte dalla compagnia TPIU con la relativa dimensione (**dimT**) e restituisca un array (di dimensione *dimResult*) con tutti i possibili collegamenti che partono da o arrivano alla città specificata, ordinati per prezzo crescente.

Il candidato scriva le opportune istruzioni nel main per testare questa funzione.

Esercizio 4 – Main (*main.c*)

Il candidato realizzi un programma in grado di:

- 1) Leggere i collegamenti (le tratte) e le prenotazioni dai rispettivi file creando due array opportunamente allocati in memoria.
- 2) Inserire una nuova prenotazione nel sistema.
- 3) Trovare tutti i treni che arrivano o partono da Bologna (codice "BOL")
- 4) Deallocare (al termine del programma) tutte le strutture allocate dinamicamente.