

Fondamenti di Informatica T1

Prova Pratica - Maggio 2014

CompitoA

Prima di cominciare: si scarichi il file **StartKitA.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 aerea FlyHigh ha deciso di realizzare un sistema informatizzato per la gestione dei voli disponibili e delle prenotazioni effettuate. Il sistema memorizza i voli offerti dalla compagnia in un opportuno file di testo *voli.txt* (presente nello start kit). Ogni riga di tale file corrisponde ad una tratta aerea tra due città (senza scali) ed è composta dai seguenti campi:

codiceVolo cittàPartenza cittàArrivo durata costo

dove codiceVolo 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 volo mentre cittàArrivo è una altra stringa di 3 caratteri che specifica la destinazione; la durata, un intero, è espressa in minuti e infine il costo è un float che indica il prezzo di un biglietto singolo per il singolo viaggio, quindi sola andata. Supponiamo 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 volo 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 voli siano presenti nel file.

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

codiceVolo nomeCliente tipo numeroBiglietti costoTotale

dove codiceVolo specifica la tratta desiderata, nomeCliente è una stringa senza spazi contenente il nome di chi ha effettuato la prenotazione (al massimo 20 caratteri), il tipo di prenotazione è un carattere che specifica se la prenotazione riguarda la sola andata ('A') oppure andata e ritorno ('R'), il numeroBiglietti è un intero che indica quanti biglietti sono stati prenotati e il costoTotale è un float con il prezzo totale pagato dal cliente per la sua prenotazione. Anche in questo caso non è noto a priori il numero di prenotazioni presenti nel file.

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

Si definisca un'opportuna struttura dati **Volo**, al fine di rappresentare i dati registrati nel file *voli.txt* come specificato sopra, e si realizzi una procedura:

Volo* leggiVoli(char fileName[], int *dim);

Fondamenti di Informatica T1

Prova Pratica - Maggio 2014

CompitoA

che, ricevuto in ingresso il nome di un file contenente i dati dei voli, 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 voli letti) in **dim**. Come già detto, il numero di voli presenti nel file non è noto a priori occorre determinarlo per allocare la memoria necessaria.

Si realizzi in seguito la procedura

void stampaVolo(Volo v);

che, ricevuto in ingresso un Volo 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).

Esercizio 2 – Inserimento prenotazione (gestioneVoli .h/gestioneVoli .c)

Si realizzi una funzione

int inserisciPrenotazione(FILE* fp, Prenotazione *p, int dimP, Volo* v, int dimV);

che preso in ingresso un puntatore ad un file di testo già aperto, un array di **Prenotazione** p e un array di **Volo** v, con le relative dimensioni (rispettivamente dimP e dimV) provveda a scrivere nel file passato come argomento la nuova prenotazione inserita dall'utente.

Tramite tastiera, l'utente dovrà inserire 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 **v**, in caso positivo dovrà poi calcolare il prezzo totale della prenotazione, tenendo conto del numero di biglietti richiesti e del fatto che il volo 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 1.6.

La funzione deve ritornare il valore '0' se l'inserimento è andato a buon fine, '-1' in caso

Fondamenti di Informatica T1
Prova Pratica - Maggio 2014
CompitoA

contrario.

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

Esercizio 3 – Trova voli (gestioneVoli .h/gestioneVoli .c)

Il candidato realizzi una funzione

Volo* trovaVoli(char* citta, Volo* v, int dimV, int* dimResult);

che prenda in ingresso il codice di una città, l'array contenente tutti i voli offerti dalla compagnia area con la relativa dimensione (**dimV**) e restituisca un array con tutti i possibili voli che partono o arrivano alla città specificata, ordinati per prezzo crescente; utilizzare uno degli algoritmi di ordinamento visti a lezione.

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 voli e le prenotazioni dai rispettivi file creando due array.
- 2) Inserire una nuova prenotazione nel sistema.
- 3) Trovi tutti i voli che atterrano a Bologna (codice "BLQ")
- 4) Deallocare (al termine del programma) tutte le strutture allocate dinamicamente.