

Fondamenti di Informatica T1  
Prova Pratica - Settembre 2013  
Compito

**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*.

Un'agenzia di viaggi ha deciso di gestire l'offerta di pacchetti vacanze tramite un sistema informatizzato. Tramite questo sistema i clienti possono effettuare le loro richieste di viaggio e prenotarle in modo automatizzato.

In un file di testo, denominato "*pacchetti\_vacanze.txt*", viene salvato l'elenco delle vacanze disponibili. Ogni riga corrisponde ad un pacchetto vacanza ed è formata dai seguenti campi:

**codice destinazione tipologia prezzo posti\_disponibili**

dove il codice è una stringa di 4 caratteri alfanumerici che identifica il pacchetto vacanza, la destinazione è una stringa (senza spazi) di al massimo 50 caratteri, la tipologia di prodotto è una stringa senza spazi (le categorie presenti al momento sono "*Relax*", "*Crociera*", "*Avventura*" e "*Cultura*"), il prezzo è un float che rappresenta il costo in euro per persona per la vacanza, il numero di posti disponibili indica quanti clienti possono ancora effettuare una prenotazione prima che l'offerta sia esaurita.

In un secondo file "*prenotazioni.txt*" vengono salvate le richieste di viaggio fatte dai clienti dell'agenzia viaggi; supponiamo per semplicità che nel file sia presente al massimo una prenotazione per cliente. In ogni riga del file sono memorizzati:

**nome\_cliente destinazione\_desiderata numero**

dove il nome del cliente è una stringa senza spazi, la destinazione indica la meta desiderata dal cliente e il numero è un intero che indica il numero di posti che il cliente ha intenzione di prenotare.

**Esercizio 1 - Lettura da file (*agenzia\_vacanze.h/agenzia\_vacanze.c*)**

Si definisca un'opportuna struttura dati **Vacanza** al fine di rappresentare i pacchetti vacanza offerti in vendita, in particolare tenendo traccia di codice identificativo del prodotto, nome, tipologia e prezzo.

Il candidato realizzi poi la funzione:

**Vacanza \* leggiVacanze (char \* nomefile, int \* dimV);**

che, ricevuto come parametro di ingresso il nome di un file, provveda ad aprirlo e ne legga il contenuto, ovvero le vacanze offerte. Si noti che non è noto a priori quanti pacchetti vacanze siano registrati nel file, dunque sarà necessario determinarne il

# Fondamenti di Informatica T1

## Prova Pratica - Settembre 2013

### Compito

numero e poi allocare memoria dinamicamente nella dimensione opportuna. Tale funzione deve restituire la dimensione logica dell'array in **dimV**.

Procedendo in maniera simile al punto precedente, definire un'opportuna struttura dati **Prenotazione** al fine di rappresentare le prenotazioni fatte dai clienti, in particolare considerando nome del cliente, destinazione desiderata e posti richiesti.

Realizzare quindi la funzione:

**Prenotazione \* leggiPrenotazioni (char \* nomefile, int \* dimP);**

che provveda a leggere da un file di testo (il cui nome è passato come parametro di ingresso) le prenotazioni ricevute; anche in questo caso non è dato sapere a priori il numero di righe presenti nel file di testo e quindi sarà necessario allocare dinamicamente l'array di strutture **Prenotazione**. Nuovamente, restituire la dimensione logica del vettore tramite il parametro **dimP**.

Il candidato abbia cura di realizzare nel main opportuni test al fine di verificare il corretto funzionamento delle funzioni di cui sopra. A tale scopo, si utilizzi il file di testo "*pacchetti\_vacanze.txt*", contenente un elenco di pacchetti vacanze, e "*prenotazioni.txt*", con le ordinazioni (presenti nello **StartKit.zip**). Una volta verificato il corretto funzionamento delle funzioni, il candidato non cancelli il codice nel main ma si limiti a commentarlo.

### Esercizio 2 – Ordinamento pacchetti vacanze (agenzia\_vacanze .h/agenzia\_vacanze.c)

Si realizzi una funzione:

**void ordina (Vacanza \* vacanze, int dim);**

che ordini i pacchetti vacanze passati come argomento; l'ordinamento deve avvenire raggruppando i pacchetti in base alla tipologia (sfruttare l'ordinamento lessicografico per decidere come ordinare le tipologie) e nel caso di pacchetti vacanze con la stessa tipologia la precedenza è data a quelli più economici.

La funzione prende anche in ingresso la dimensione del vettore da ordinare e deve anche stampare a video le vacanze correttamente ordinate.

Il candidato realizzi nel main opportuni test per verificare il corretto funzionamento della funzione.

### Esercizio 3 – Gestione prenotazioni (gestore.h/gestore.c)

Il candidato definisca una procedura:

**void gestione\_prenotazioni (Vacanza \* v, int dimV, Prenotazione \* p, int dimP);**

che preso in ingresso un vettore **v** di strutture dati **Vacanza** e un vettore **p** di strutture dati **Prenotazione** (con le rispettive dimensioni logiche), tenti di soddisfare tutte le richieste fatte dai clienti. Ciò implica che per ogni prenotazione in **p** la funzione dovrà verificare la disponibilità di posti per il corrispondente pacchetto vacanza in **v** e dovrà

# Fondamenti di Informatica T1

## Prova Pratica - Settembre 2013

### Compito

anche controllare che la destinazione desiderata dal cliente sia effettivamente compresa all'interno di un pacchetto vacanza offerto.

Per ogni prenotazione possono verificarsi tre casi distinti:

- la prenotazione del cliente è valida (la destinazione desiderata è servita da almeno un pacchetto vacanza) e ci sono abbastanza posti disponibili – in questo caso occorre modificare il vettore delle vacanze **v**, aggiornando il numero di posti rimasti, e poi stampare a video il dettaglio della vacanza selezionata e del cliente, calcolando anche il prezzo totale della prenotazione;
- la prenotazione del cliente è valida ma non ci sono abbastanza posti disponibili – in questo caso non è necessario modificare il vettore **v**, ma occorre stampare a video le possibili alternative per il cliente, ovvero i pacchetti vacanze che rientrano nella stessa tipologia della prenotazione originale, senza preoccuparsi del numero di posti disponibili (supponiamo che tra i pacchetti vacanze offerti ce ne siano sempre almeno due per ogni categoria di vacanza);
- la prenotazione non è valida – in questo caso stampare a video un codice di errore e una stringa che spieghi che non sono ancora previste vacanze presso la destinazione richiesta dal cliente.

Si realizzino nel main le opportune istruzioni per verificare il corretto funzionamento delle funzioni implementate.

#### **Esercizio 4 – Main (main.c)**

Il candidato realizzi un programma in grado di:

- 1) Leggere i pacchetti viaggio offerte dall'agenzia e le prenotazioni dei clienti dai rispettivi file creando due array.
- 2) Ordinare l'array relativo ai pacchetti vacanze.
- 3) Gestire le prenotazioni dei clienti tramite la funzione vista al punto 3.
- 4) Deallocare (al termine del programma) tutte le strutture allocate dinamicamente.