

Fondamenti di Informatica T1  
Prova Pratica – 11 Febbraio 2015  
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**). 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*.

Il parco divertimenti Divertilandia vuole un nuovo software per gestire le code delle proprie attrazioni. Il software gestisce le code delle attrazioni in modo tale che ogni famiglia possa entrare nell'attrazione solo se ci sono abbastanza posti per ospitarla per intero, inoltre sono presenti due tipi di famiglie: standard e premium, a fronte di una sovrattassa, una famiglia entra come premium e ha accesso prioritario a tutte le attrazioni.

All'interno del file "famiglie.txt" sono memorizzati i dati delle famiglie nel parco, nel seguente formato:

**codiceFamiglia numeroComponenti tipologia**

Dove codiceFamiglia è una stringa di 5 caratteri che identifica univocamente la famiglia, numeroComponenti è un intero e indica il numero di componenti della famiglia e tipologia è un carattere che può assumere il valore 'S' per le famiglie standard e 'P' per le famiglie premium

Il file "code.txt" contiene lo stato delle code di tutte le attrazioni, il file è nel seguente formato:

**idAttrazione codiceFamiglia**

Come nel file precedente idAttrazione è un intero che identifica l'attrazione e codiceFamiglia è una stringa di 5 caratteri che identifica la famiglia in coda; si assuma che l'ordine di arrivo delle famiglie in coda coincida con l'ordine degli inserimenti nel file.

Ogni attrazione può ospitare lo stesso numero massimo di persone, questo numero è 15; in ogni coda possono essere presenti al massimo 30 persone.

Si definiscano le costanti **NUMERO\_POSTI** e **LUNGHEZZA\_CODA** contenenti rispettivamente il numero massimo di persone che possono entrare in un attrazione (15) e il numero massimo di persone che possono essere presenti in una coda (30).

**Esercizio 1 - Lettura da file (divertilandia.h/ divertilandia.c)**

Si definisca un'opportuna struttura dati **Famiglia** al fine di rappresentare le famiglie nel parco e tenente traccia di tutte le informazioni relative ad essa.

Il candidato realizzi poi la funzione:

Fondamenti di Informatica T1  
Prova Pratica – 11 Febbraio 2015  
Compito

**Famiglia \* leggiFamiglia (char\* nomefile, int\* numF)**

Che, ricevuto come parametro di ingresso il nome di un file, provveda ad aprirlo e ne legga il contenuto, ovvero la lista delle famiglie. Non è noto a priori quante famiglie siano nel parco, quindi sarà necessario allocare memoria dinamicamente nella dimensione opportuna. Tale funzione deve restituire la dimensione logica dell'array in **numF**.  
Si realizzi inoltre una procedura:

**void stampaFamiglia (Famiglia f)**

Che stampi a video (in maniera leggibile – usare spazi, separatori, fine linea, etc.) le informazioni relative ad una singola famiglia presa in ingresso dalla procedura.

Si definisca la struttura **Coda** in modo da contenere i dati delle code.  
Realizzare quindi la funzione:

**Coda \* leggiCoda (char\* nomefile, int\* dimC)**

Che provveda a leggere da un file di testo (il cui nome è passato come parametro di ingresso) le famiglie in coda; 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 **Coda**. Nuovamente, restituire la dimensione logica del vettore tramite il parametro **dimC**.

Analogamente al caso precedente, definire una procedura che stampi a video tutte le informazioni relative alla coda:

**void stampaCoda (Coda c)**

Il candidato abbia cura di realizzare nel main opportuni test al fine di verificare il corretto funzionamento delle funzioni di cui sopra, sfruttando i file di testo "*famiglie.txt*" e "*code.txt*" (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 – Inserimento in coda (divertilandia.h/ divertilandia.c)**

Si definisca la procedura

**Int contaPersoneInCoda(int idAttrazione, Famiglia \* f, int numF, Coda \* c, int dimC)**

Che preso in ingresso l'identificativo dell'attrazione, un vettore **f** di strutture dati **Famiglia** con la sua dimensione logica e un vettore **c** di strutture dati **Coda** con la sua dimensione logica, cerchi tutte le famiglie in coda sull'attrazione in questione e ritorni il totale di persone in coda (si utilizzi il campo **numeroComponenti** della struttura **Famiglia**). Se l'id dell'attrazione non viene trovato negli elementi del vettore **c**, la funzione deve restituire '-1'.

Si definisca la procedura

**Coda \* inserisciFamigliaInCoda(Famiglia \* f, int numF, Coda \* c, int dimC)**

Che preso in ingresso un vettore **f** di strutture dati **Famiglia** con la sua dimensione logica e un vettore **c** di strutture dati **Coda** con la sua dimensione logica, consenta di inserire una famiglia nella coda dell'attrazione.

# Fondamenti di Informatica T1

## Prova Pratica – 11 Febbraio 2015

### Compito

La funzione chiederà di inserire da tastiera il codice della famiglia (già tra le famiglie presenti) e l'id dell'attrazione, quindi conterà il numero di persone in coda su quella attrazione (si utilizzi la funzione **contaPersonelnCoda**), se il numero di persone in coda consente di ospitare anche la nuova famiglia senza eccedere la lunghezza massima (**LUNGHEZZA\_CODA**), allora verrà creato dinamicamente un nuovo vettore di strutture **Coda** della dimensione esatta per contenere, oltre alle famiglie già in coda, anche la famiglia appena inserita, quindi il nuovo vettore verrà restituito; se invece la dimensione massima della coda dell'attrazione non permettesse di ospitare la famiglia oppure se il codice della famiglia inserito non fosse presente nel vettore di strutture **Famiglia** oppure se l'id dell'attrazione non venisse trovato dalla funzione **contaPersonelnCoda**, verrà restituito il vecchio vettore di strutture **Coda**.

Si realizzino nel main le opportune istruzioni per verificare il corretto funzionamento della funzione.

#### Esercizio 3 – Apertura cancelli ([divertilandia.h](#)/[divertilandia.c](#))

Si realizzi una funzione:

**Coda \* apriCancelli (Famiglia \* f, int numF , Coda \* c, int dimC)**

Che preso in ingresso un vettore **f** di strutture dati **Famiglia**, con la sua dimensione logica e un vettore **c** di strutture dati **Coda** con la sua dimensione logica, richieda l'inserimento da tastiera di un **idAttrazione** per il quale verranno aperti i cancelli, quindi riempirà l'attrazione togliendo le famiglie dalla coda dando prima la precedenza alle famiglie Premium e solo successivamente valutando le famiglie Standard, la funzione deve stampare a video ogni famiglia che entra nell'attrazione e terminare quando nessuna famiglia può più entrare nell'attrazione senza eccedere il numero di posti (**NUMERO\_POSTI**) oppure nel caso in cui non siano presenti ulteriori famiglie in coda. La funzione deve inoltre ritornare un nuovo vettore delle code contenente tutti gli elementi della coda, ad eccezione delle famiglie appena entrate nell'attrazione; se l'id dell'attrazione non viene trovato tra quelli presenti nella coda, allora verrà restituito il vettore **c** preso in ingresso.

Come sempre, si realizzino nel main le opportune istruzioni per verificare il corretto funzionamento della funzione.

#### Esercizio 4 – Salva coda ([divertilandia.h](#)/[divertilandia.c](#))

Si definisca la funzione

**Int salvaCode(char\* nomefile ,Coda \*c)**

La funzione prende in input una stringa contenente il nome del file e un vettore **c** di strutture **Coda**, si apra in scrittura il file **nomeFile** e si scrivano i dati di tutte le code, la funzione deve restituire '0' in caso di corretto funzionamento, '-1' altrimenti.

Fondamenti di Informatica T1  
Prova Pratica – 11 Febbraio 2015  
Compito

Si realizzino nel main le opportune istruzioni per verificare il corretto funzionamento della funzione, scrivendo nel file *"code.txt"*.

**Esercizio 5 – Main (*main.c*)**

Il candidato realizzi un programma in grado di:

- 1) Leggere famiglie e code dai rispettivi file creando due array.
- 2) Mostrare a video i due vettori appena creati.
- 3) Inserire una famiglia in coda.
- 4) Mostrare a video le code.
- 5) Aprire i cancelli di un attrazione.
- 6) Salvare le code su file.
- 7) Deallocare (al termine del programma) tutte le strutture allocate dinamicamente.