

Fondamenti di Informatica e Laboratorio T-AB  
Prova Pratica - 10 Gennaio 2012  
Compito A

**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 trasmissione televisiva SuperGear ha in programma una puntata speciale che prevede tre sfide per diverse supercar. In particolare i tecnici del programma memorizzano tutti i dati delle vetture in un file di nome **cars.txt**. Ogni auto in gara è rappresentata tramite una riga composta dal codice auto (5 caratteri), il nome (stringa senza spazi, dimensione massima 30), la dimensione cc del motore (int), la potenza in Hp (int) e l'anno di produzione del modello (int). Ad esempio

**CAR01 Lamborghini\_Miura 3900 350 1966**

In un secondo file, di nome **sfide.txt**, sono memorizzati i tempi delle misurazioni delle tre sfide (A,B e C) per ogni auto. Attualmente non tutte le misurazioni sono state effettuate. Il file presenta, su ogni riga, una misurazione:

**A CAR01 3,56**

caratterizzata dal codice sfida (A,B o C), dal codice auto e dal tempo (double o float). Se l'auto non ha ancora affrontato la sfida la misurazione non è presente nel file, mentre se l'auto **NON** ha superato la sfida (ad esempio perché è uscita di strada) la misurazione è un numero negativo.

### *Esercizio 1 – lettura dei dati (cars.h/cars.c)*

Il candidato realizzi un modulo che definisca una struttura dati opportuna, di nome **Supercar**, per rappresentare i dati relativi ad un'auto e le sue tre sfide, e definisca le funzioni per (a) leggere dal file **cars.txt** l'elenco delle auto con i dati relativi e (b) leggere dal file **sfide.txt** le misurazioni delle sfide. NB: La struttura dati **Supercar** deve contenere sia i dati tecnici letti dalla funzione (a), sia le misurazioni delle sfide (**A,B e C**), lette tramite la procedura (b).

- a) La funzione di lettura riceve come parametri di ingresso il nome di un file (da cui andare a leggere) e deve restituire un array di **Supercar** allocato dinamicamente (si abbia cura di allocare solo ed unicamente lo spazio strettamente necessario); inoltre la dimensione del vettore deve essere restituita tramite un ulteriore apposito parametro in ingresso. Questa prima funzione deve assegnare ai campi relativi alle sfide A, B e C il valore 0.

**Supercar \* leggiAuto(char \* nomeFile, int \* dim);**

- b) La procedura di ricerca riceve come parametri di ingresso il nome di un file (da cui andare a leggere) e l'array di strutture dati **Supercar** (creato dalla funzione a) con la sua dimensione;

**void leggiMisurazioni(char \* nomeFile, Supercar \* s, int dim);**

**NB:** le misurazioni vanno memorizzate nelle strutture dati **Supercar** (nei campi relativi alle sfide) dell'array passato come parametro. Nel caso in cui un codice auto letto dal file **sfide.txt** **NON** trovasse corrispondenza tra le auto memorizzate nel vettore di strutture **Supercar**, si ignori la misurazione e si stampi a video un messaggio di errore relativo a quel codice.

Fondamenti di Informatica e Laboratorio T-AB  
Prova Pratica - 10 Gennaio 2012  
Compito A

Contestualmente, il candidato scriva nel main opportune istruzioni per invocare le funzioni definite al fine di verificarne la corretta implementazione. Una volta verificato il corretto funzionamento delle funzioni, il candidato non cancelli il codice nel main ma si limiti a commentarlo.

**Esercizio 2 – Gestione delle sfide (funzioni.h/funzioni.c)**

Il candidato definisca un'opportuna funzione che prenda in ingresso l'array di **Supercar**, la relativa dimensione ed un carattere identificativo della sfida (A,B o C).

Tale funzione deve verificare quante macchine hanno superato la sfida (che hanno cioè valori positivi) e creare dinamicamente un nuovo array contenente le sole auto che hanno superato la sfida ordinate in ordine crescente per misurazione. Il vettore creato deve essere restituito dalla funzione; inoltre la dimensione del vettore deve essere restituita tramite un ulteriore apposito parametro passato per riferimento.

```
Supercar * sfida(char sfida, Supercar * Auto, int dimAuto, int * dim);
```

Come suggerimento si consiglia al candidato di scrivere una funzione dedicata per ordinare il vettore **Supercar**; a tal fine si usi un algoritmo visto a lezione opportunamente modificato. Alternativamente si può creare il vettore inserendo ordinatamente le strutture.

Contestualmente, il candidato scriva nel main le istruzioni per invocare le funzioni definite per tutte e tre le sfide. Quindi stampi a video, per ogni sfida, la lista ordinata delle auto (è sufficiente stampare il nome auto e la misurazione relativa alla sfida).

**Esercizio 3 – Inserimento misurazione (funzioni.h/funzioni.c)**

Il candidato definisca una procedura che aggiunga ad un file (il cui nome viene passato come parametro) una riga corrispondente ad una misurazione.

```
void inserisciMisurazione(char * nomeFile, char * codiceAuto, double mis);
```

Tale funzione prende come parametri il codice dell'auto ed il valore (double) della misurazione. Il tipo di sfida deve essere chiesto all'utente prima dell'inserimento nel file.

Contestualmente, il candidato scriva nel main opportune istruzioni per invocare la funzione definita al fine di verificarne la corretta implementazione.

**Esercizio 4 – Media (funzioni.h/funzioni.c)**

Si realizzi una procedura che, ricevuti in ingresso un array di strutture dati **Supercar** e la relativa dimensione, calcoli la media matematica dei tempi delle misurazioni. Tale valore deve essere stampato a video e deve essere ritornato come valore double.

```
double media(Supercar * Auto, int dimAuto);
```

**Esercizio 5 – Main (main.c)**

Il candidato realizzi un programma **main.c** che crei un array di strutture dati **Supercar** caricando i dati dai file compresi nello start kit (cars.txt e sfide.txt). In seguito, tramite la funzione definita nell'esercizio 2, il programma deve creare un array ordinato per ogni tipo di sfida (A,B, o C). Per ogni array, tramite la funzione **media**, va calcolata la media. Infine il candidato inserisca la seguente misurazione nel file sfide.txt:

**C CAR13 3,36**

Il candidato abbia cura di deallocare (al termine del programma) tutte le strutture allocate dinamicamente.