

Fondamenti di Informatica e Laboratorio T-AB Simulazione della Prova d'Esame - 28 Novembre 2008

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 sorgenti 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 Società "Autostrade Nevose" sta mettendo a punto un nuovo tipo di autovelox, che registra gli istanti temporali in cui un'auto passa in due punti diversi. Lo scopo è multare quelle autovetture che percorrono la distanza tra i due punti con una velocità media superiore a quella consentita di 130 Km/h.

A tal fine la società rappresenta le informazioni in suo possesso tramite due strutture dati: nella prima struttura, di nome `rec`, vengono registrati la targa dell'auto (una stringa di 7 caratteri), gli istanti `time1` e `time2` che registrano in secondi, quando l'auto è passata sotto i rilevatori (tramite due interi), e la distanza `dist` tra i due punti di rilevazione (in Km, un float).

Tramite la seconda struttura, di nome `car`, viene registrata una targa (stringa di 7 caratteri) ed il nome del proprietario dell'auto (una stringa di al più 63 caratteri senza spazi).

Esercizio 1 - Definizione delle strutture dati (file main.c)

Si definiscano in maniera opportuna le strutture dati `rec` e `car` atte a contenere le informazioni riguardo le auto ed i transiti sotto i rilevatori.

Esercizio 2 - Calcolo della velocità media (file main.c)

Si realizzi la seguente funzione:

```
float velocitaMedia(rec r);
```

che, ricevuti in ingresso una struttura dati di tipo `rec`, calcoli la velocità media del veicolo. Poiché gli istanti di rilevazione sono espressi in secondi, rispetto a quando il sistema è stato messo in funzione per la prima volta, la velocità media in Km/h è data dalla seguente formula:

$$v = (\text{dist} / (\text{time2} - \text{time1})) * 3600$$

Esercizio 3 - Selezione delle autovetture multabili (file main.c)

Si implementi la seguente funzione:

```
int multabili(rec src[], int dim1, rec dest[], int dim2);
```

che, ricevuti in ingresso un array `src` di strutture `rec` con la sua dimensione `dim1`, ed un array `dest` con la sua dimensione fisica `dim2`, copi in `dest` tutte le strutture dati di `src` per le quali risulti una velocità media superiore ai 130 Km/h. La funzione deve restituire il numero di elementi copiati in `dest`, ovvero la sua dimensione logica.

Esercizio 4 - Lettura da un file delle strutture rec (file main.c)

Si realizzi la funzione

```
int leggiRec(FILE * fp, rec dest[], int dim);
```

che riceva in ingresso un puntatore ad un file già opportunamente aperto, un array `dest` di tipo `rec` e la sua dimensione fisica. La funzione deve provvedere a leggere le strutture dati presenti nel file, e a memorizzarle nell'array `dest`. Si supponga che il file, di nome `recs.txt`, sia un file di testo dove, per ogni riga, sono memorizzate rispettivamente la targa, i valori `time1` e `time2`, e la distanza tra ogni rilevatore. Il candidato usi il file `recs.txt` in dotazione come esempio ed ai fini di debug della funzione. La funzione deve restituire il numero di elementi copiati in `dest`, ovvero la sua dimensione logica.

Fondamenti di Informatica e Laboratorio T-AB Simulazione della Prova d'Esame - 28 Novembre 2008

Esercizio 5 - Lettura da un file delle strutture car (file main.c)

Si realizzi la funzione

```
int leggiCar(FILE * fp, car dest[], int dim);
```

che riceve in ingresso un puntatore ad un file già opportunamente aperto, un array `dest` di tipo `car`, e la sua dimensione fisica. La funzione deve leggere dal file specificato le strutture dati di tipo `car` e memorizzarle nel vettore `dest`; la funzione deve restituire il numero di elementi letti, ovvero la dimensione logica del vettore `dest`.

A tal fine si supponga che le informazioni nel file siano memorizzate col seguente metodo: in ogni riga viene messa prima la targa dell'auto e poi, separata da un spazio, il nome del proprietario dell'auto. Il candidato usi il file `cars.txt` in dotazione al fine di provare ed in fase di debug della funzione.

Esercizio 6 - Selezione di una struttura dati di tipo car (file main.c)

Si realizzi la funzione

```
int getCar(char * targa, car src[], int dim, car * dest);
```

che riceve in ingresso una stringa di nome `targa`, un array di strutture dati di tipo `car` assieme alla dimensione di tale array, ed una struttura dati `dest` di tipo `car` passata per riferimento. La funzione deve copiare in `dest` la struttura dati `car` di `src` la cui targa è uguale a quella specificata come parametro d'ingresso. La funzione deve restituire 1 se la targa è stata trovata, 0 altrimenti.

Esercizio 7 - Stampa dei nomi di chi ha violato il limite di velocità (file main.c)

Si realizzi un programma main che compia le seguenti operazioni:

- Legga dal file `recs.txt` i dati relativi ai rilevamenti di velocità, utilizzando la funzione di cui al punto 4. Si supponga per semplicità che il file contenga al massimo 10 strutture dati.
- Legga dal file `cars.txt` l'elenco delle strutture dati car relative alle targhe delle auto ed ai loro proprietari. Si supponga per semplicità che il file contenga al massimo 10 strutture dati. A tal fine si usi la funzione di cui al punto 5.
- Selezioni in un apposito vettore le strutture dati relative ad infrazioni del limite di velocità. A tal fine si usi la funzione di cui al punto 3.
- Stampi a video i nomi di coloro che hanno infranto il limite di velocità, con anche la velocità media registrata. A tal scopo si "incrocino" opportunamente gli array letti ai punti precedenti, e si usi la funzione di cui al punto 6.

Fondamenti di Informatica e Laboratorio T-AB
Simulazione della Prova d'Esame - 28 Novembre 2008

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define DIM 10

typedef struct {
    char targa[8];
    char nome[64];
} car;

typedef struct {
    char targa[8];
    int time1;
    int time2;
    float dist;
} rec;

float velocitaMedia(rec r) {
    return (r.dist / (r.time2 - r.time1))*3600;
}

int multabili(rec src[], int dim1, rec dest[], int dim2) {
    int i = 0;
    int result = 0;

    while (i < dim1 && result < dim2) {
        if (velocitaMedia(src[i]) > 130) {
            dest[result] = src[i];
            result++;
        }
        i++;
    }
    return result;
}

int leggiRec(FILE * fp, rec dest[], int dim) {
    int result = 0;

    while (fscanf(fp, "%s %d %d %f",
        dest[result].targa, &dest[result].time1,
        &dest[result].time2, &dest[result].dist) == 4
        && result < dim)
        result++;
    return result;
}

int leggiCar(FILE * fp, car dest[], int dim) {
    int result = 0;

    while (fscanf(fp, "%s %s\n", dest[result].targa, dest[result].nome) == 2
        && result < dim)
        result++;
    return result;
}
```

Fondamenti di Informatica e Laboratorio T-AB
Simulazione della Prova d'Esame - 28 Novembre 2008

```
car getCar(char * targa, car src[], int dim, car * dest) {
    int i = 0;
    int trovato = 0;

    while (i < dim && !trovato) {
        if (strcmp(targa, src[i].targa) == 0) {
            trovato = 1;
            *dest = src[i];
        }
        else
            i++;
    }
    return trovato;
}

int main(void) {
    FILE * fp1, * fp2;
    rec r[DIM];
    car c[DIM];
    rec m[DIM];
    int i;
    int dim_c, dim_r, dim_m;
    car temp;

    fp1 = fopen("cars.txt", "r");
    fp2 = fopen("recs.txt", "r");

    dim_c = leggiCar(fp1, c, DIM);
    for (i=0; i < dim_c; i++)
        printf("%s %s\n", c[i].targa, c[i].nome);

    dim_r = leggiRec(fp2, r, DIM);
    for (i=0; i < dim_r; i++)
        printf("%s %d %d %f\n", r[i].targa, r[i].time1, r[i].time2,
            r[i].dist);

    dim_m = multabili(r, dim_r, m, DIM);
    for (i=0; i < dim_m; i++) {
        printf("%s %d %d %f\n", m[i].targa, m[i].time1, m[i].time2,
            m[i].dist);
        printf("Velocita: %f\n", velocitaMedia(m[i]));
    }

    for (i=0; i < dim_m; i++) {
        for (i=0; i < dim_m; i++) {
            if (getCar(m[i].targa, c, dim_c, &temp))
                printf("%s viaggiava a %f Km/h\n",
                    temp.nome, velocitaMedia(m[i]));
        }
    }

    fclose(fp1);
    fclose(fp2);
    system("PAUSE");

    return (0);
}
```