

Fondamenti di Informatica e Laboratorio T-AB
Prova Pratica – 17 Giugno 2013
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.

La torrefazione EFFECAFFÈ utilizza un sistema informatico per gestire la consegna di ordini ai suoi punti vendita sul territorio. In particolare, le informazioni sulle disponibilità dei prodotti sono memorizzate nel file di testo "disponibilita.txt" (vedi startkit), le cui righe sono strutturate come segue:

codiceMagazzino codiceProdotto pos_x pos_y prezzo

dove "codiceMagazzino" e "codiceProdotto" sono due stringhe rispettivamente di 4 e 5 caratteri, "pos_x" e "pos_y" sono due interi che rappresentano la posizione del magazzino (come coordinate x/y). Infine, "prezzo" è un float che indica il prezzo unitario del prodotto. Le informazioni sugli ordini sono invece memorizzate nel file di testo "ordini.txt" (vedi startkit), le cui righe sono strutturate come segue:

codiceNegozio codiceProdotto pos_x pos_y quantita

dove "codiceNegozio" è un stringa di 4 caratteri, "codiceProdotto" è come sopra, "pos_x" e "pos_y" sono due interi che rappresentano la posizione del negozio. Infine, "quantita" è un intero che indica la quantità del prodotto che è stata ordinata.

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

A) Si definisca una struttura dati per rappresentare le informazioni sulle disponibilità dei prodotti e si realizzi una funzione:

Disponibilita* leggiDisponibilita(char* fileName, int* dim);

che, ricevuto in ingresso il nome di un file contenente i dati sulle disponibilità, legga tali informazioni e le memorizzi in un array allocato dinamicamente. Tale funzione deve restituire un riferimento all'area di memoria allocata (array) e la sua dimensione logica (numero di prodotti letti) in **dim**. Il numero di disponibilità nel file non è noto a priori ed occorre determinarlo per poter allocare la memoria necessaria.

Si realizzi in seguito la funzione:

void stampaDisponibilita(Disponibilita d);

che, ricevuta in ingresso una **Disponibilita**, ne stampi a video il contenuto.

B) Si definisca poi una struttura dati per memorizzare le informazioni sugli ordini e si realizzi la funzione:

Ordine* leggiOrdini(char* fileName, int* dim);

che, ricevuto in ingresso il nome di un file contenente i dati sugli ordini, legga tali informazioni e le memorizzi in un array allocato dinamicamente. Tale funzione deve restituire un riferimento all'area di memoria allocata (array) e la sua dimensione logica (numero di opinioni lette) in **dim**.

Si realizzi in seguito la funzione:

void stampaOrdine(Ordine o);

che, ricevuto in ingresso un **Ordine**, ne stampi a video il contenuto.

C) Si realizzino nel main le opportune istruzioni per leggere e stampare a video tutte le informazioni contenute nei due files "disponibilita.txt" e "ordini.txt".

Fondamenti di Informatica e Laboratorio T-AB
Prova Pratica – 17 Giugno 2013
Compito A

Esercizio 2 – Determinazione della disponibilità più vicina (funzioni.h/funzioni.c)

A) Si scriva la funzione

float distanza_quadra(Ordine o, Disponibilita d);

Che dato un **Ordine** ed una **Disponibilita**, restituisca il quadrato della loro distanza euclidea (ossia $(x_o - x_d)^2 + (y_o - y_d)^2$).

B) Si utilizzi poi tale funzione per definire:

Disponibilita piu_vicina(Ordine o, Disponibilita* disp, int ndisp);

che, dato un **Ordine** "o" ed un array di **Disponibilita** "disp" (di dimensione "ndisp"), individua la disponibilità più vicina relativa al prodotto richiesto nell'ordine. La funzione restituisce la **Disponibilita** più vicina.

C) Si estenda il main in modo da stampare, per ogni ordine, le informazioni relative alla disponibilità più vicina ad esso.

Esercizio 3 – Controllo di validità degli ordini (funzioni.h/funzioni.c)

A) Si realizzi poi la funzione:

int controlla(char* codiceNegozio, char* codiceMagazzino, Ordine* o, int no, Disponibilita* d, int nd);

che, date le informazioni sugli ordini e sulle disponibilità (array "o" e "d", di dimensione rispettivamente "no" ed "nd"), verifichi la validità di una coppia negozio/magazzino. La coppia è valida se i prodotti negli ordini relativi al negozio (specificato da "codiceNegozio") sono disponibili nel magazzino (specificato da "codiceMagazzino"), ossia se per ognuno di tali prodotti esiste almeno una disponibilità nel magazzino. Tale funzione dovrà restituire 1 se la coppia è valida, 0 altrimenti.

B) Si estenda il main in modo da verificare la validità delle seguenti coppie negozio/magazzino: N001/M001, N001/M003, N003/M003.

Esercizio 4 – Estrazione ed ordinamento di disponibilità (funzioni.h/funzioni.c)

A) Si sviluppi la funzione:

Disponibilita* ottieni_ordinate(char* codiceProdotto, Disponibilita* d, int nd, int* nd2);

Che permette di ottenere dall'array di **Disponibilita** "d", di dimensione "nd", le disponibilità relative al prodotto di codice "codiceProdotto". Tali disponibilità dovranno essere memorizzate in un array allocato dinamicamente ed ordinate per prezzo crescente (si consiglia l'utilizzo dell'algoritmo di ordinamento *insert sort* modificato opportunamente). La funzione restituisce un puntatore all'area di memoria allocata (l'array) e la sua dimensione logica (mediante il parametro "nd2").

B) Si estenda il main per ottenere e stampare a video l'array ordinato delle disponibilità per il prodotto avente codice "CFF01".