

Fondamenti di Informatica e Laboratorio T-AB ETA
Prova Pratica - 11 Luglio 2011
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 **codice** 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*.

L'amministratore G. Neri utilizza un sistema informatizzato per la gestione dei condomini. In particolare, l'amministratore salva i dati generici dei condomini nel file testuale "*condomini.txt*". Per ogni condominio vengono salvati il codice del condominio (valore intero), il CAP (intero di 5 cifre), il nome della via (stringa senza spazi di 50 caratteri), ed il numero civico (int); ad esempio:

34 40133 via_risorgimento 3

In un altro documento sono memorizzate tutte le comunicazioni inerenti ai condomini; tali comunicazioni sono di due tipi: (A) comunicazioni generiche, (E) comunicazioni economiche. Ogni tipologia ha una struttura differente, ma occupa una sola riga del documento. La prima tipologia è strutturata come segue:

A 34 questo spazio è dedicato alla comunicazione

dove il primo carattere contraddistingue la tipologia, il valore numerico è il codice condominio ed in seguito, fino a fine riga, è presente una stringa **CON SPAZI** (massimo 100 caratteri). La seconda tipologia (E) è definita così:

E 34 2454 caldaia 2010 7 11

dove i primi elementi si riferiscono alla tipologia di comunicazione ed al codice condominio, il terzo è un valore intero rappresentante un importo monetario lordo; in seguito è presente una stringa di descrizione massimo 50 caratteri **SENZA SPAZI** ed infine tre interi rappresentanti una data con il formato anno mese giorno.

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

Si definisca un'opportuna struttura dati, al fine di rappresentare i dati registrati nel file "*condomini.txt*", come specificato sopra, e si realizzi una funzione:

Condomino* readCondomini(char* fileName, int *dim)

che, ricevuto in ingresso il nome di un file contenente i dati dei condomini, legga i dati relativi, allochi la quantità di memoria necessaria a contenerli e ve li memorizzi. Tale funzione deve restituire un riferimento all'area di memoria allocata (array) e la sua dimensione logica (numero di elementi letti) in **dim**. Si noti che non è noto a priori quanti condomini siano presenti nel file: sarà quindi necessario determinare quanti ve ne siano, e poi allocare la memoria necessaria dinamicamente.

Si definiscano due differenti strutture al fine di rappresentare i dati relativi ai due tipi di comunicazioni e si realizzino le funzioni di lettura per entrambe, ad esempio:

ComA* readComA(char* fileName, int *dim)

ComE* readComE(char* fileName, int *dim)

che, ricevuto in ingresso il nome di un file contenente i dati delle comunicazioni ("*comunicazioni.txt*"), legga i dati relativi e li memorizzi in un'area di memoria appositamente allocata. Tali funzioni devono restituire un riferimento all'area di memoria allocata (array) e la sua dimensione logica in **dim**.

Si realizzino nel main le opportune istruzioni per verificare il corretto funzionamento delle funzioni realizzate e si stampino a video tutti gli array creati dalle funzioni.

Fondamenti di Informatica e Laboratorio T-AB ETA
Prova Pratica - 11 Luglio 2011
Compito A

Esercizio 2 – Stampa e Ordinamento (manager.h/manager.c)

Si realizzi in seguito la procedura

ComE* stampaCondominio(Condomino *c, int dimC, ComA *a, int dimA, ComE *e, int dimE)

che riceve in ingresso l'insieme di dati dei condomini (di dimensione **dimC**) e gli elenchi di entrambe le comunicazioni **A** ed **E** (rispettivamente di dimensione **dimA** e **dimE**). Tale procedura deve chiedere all'utente un codice Condominio e stampare a video il dati generici del condominio ed, in seguito, l'elenco di tutte le comunicazioni relative al condominio. L'amministratore richiede, però, che le comunicazioni economiche E, vengano stampate in ordine temporale crescente; a tale scopo si modifichi opportunamente un algoritmo di ordinamento visto a lezione. La funzione infine deve ritornare l'elenco **di tutte** comunicazioni *tipo E* ordinate.

Il candidato scriva le opportune istruzioni nel main per testare questa funzione.

Esercizio 3 – Somma Spese (function.h/function.c)

Si realizzi inoltre una funzione

double sommaSpese(int codice, ComE *e, int dimE)

che prenda in ingresso il codice di un condominio e l'elenco delle comunicazioni E, con la relativa dimensione. Tale funzione deve sommare tutte le spese nette relative al condominio, in particolare:

- se l'ammontare lordo è inferiore a 5000 si detrae una tassazione del 10%.
- se l'ammontare supera o è uguale a 5000 si detrae una tassazione del 12%.

La funzione deve ritornare il valore (**double**) della somma calcolata.

Ad esempio, se l'ammontare fosse 100€, il valore corretto da sommare, detratte le tasse, sarebbe 90€, mentre se l'ammontare lordo fosse 10000€, il valore corretto da sommare sarebbe 8800€.

Tale funzione deve anche stampare a video la media (**double**) calcolata sui valori netti (senza tasse) delle spese relative al condominio di riferimento.

Il candidato scriva le opportune istruzioni nel main per testare questa funzione.

Esercizio 4 – Inserimento comunicazione (function.h/function.c)

Il candidato realizzi una procedura

void insert(char* fileName, char type)

che prenda in ingresso il nome dei file delle comunicazioni ("*comunicazioni.txt*") ed il tipo di comunicazione. Tale funzione deve chiedere all'utente i dati relativi ad una comunicazione (in base al tipo indicato nei parametri) ed aggiungere al file delle comunicazioni i dati necessari. In caso di comunicazione tipo E, la funzione **insert**, prima di inserire la comunicazione, deve verificare che il documento ("*comunicazioni.txt*") non contenga già una comunicazione simile: in particolare, si verifichi che non ci sia una comunicazione avente la stessa data e la stessa descrizione.

Il candidato scriva le opportune istruzioni nel main per testare questa funzione.

Suggerimento: il candidato si ricordi che per le gestione delle stringe **CON SPAZI** sono presenti apposite funzioni:

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

```
char *fgets(char *s, int size, FILE *stream);  
int fputs(const char *s, FILE *stream);
```

```
char *gets(char *s);  
int puts(const char *s);
```