Fondamenti di Informatica T-1, 2011/2012 – Modulo 2

Prova d'Esame 2A di Giovedì 12 Gennaio 2012 – tempo a disposizione 2h

Prima di cominciare: si scarichi dal sito **`http://esamix.labx`** il file **`StartKit2A.zip`** contenente i file necessari (*progetto Visual Studio* ed eventuali altri file di esempio).
Avvertenze per la consegna: 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** e i file contenuti nello StartKit.
Nota: **il main non è opzionale**; i *test* richiesti vanno implementati.
Consiglio: per verificare l'assenza di *warning*, eseguire di tanto in tanto *"Rebuild All"*.

Un club automobilistico fornisce ai soci, tramite un programma, informazioni sui distributori di carburante situati in una certa area geografica. Tali informazioni sono memorizzate in un file di testo, di nome "**`distributori.txt`**", dove per ogni linea del file di testo sono memorizzate le informazioni relative a un singolo distributore. In particolare, per ogni riga del file, sono memorizzate: la **provincia** in cui è situato il distributore (una stringa di due caratteri, senza spazi), il **nome** del distributore (una stringa di al più 255 caratteri, senza spazi), il **prezzo della benzina** verde (un intero indicante il costo in centesimi di euro) e il **prezzo del gasolio** (ancora un intero indicante il prezzo in centesimi); ognuno di questi campi è separato tramite uno spazio, ed inoltre non è noto a priori quanti distributori siano memorizzati nel file di testo.
In un secondo file di testo "**`regioni.txt`**" è contenuto l'elenco delle province e delle regioni a cui queste province appartengono. In particolare, in ogni riga del file è contenuto il nome della regione (una stringa di al più 255 caratteri, senza spazi) e la sigla della provincia (una stringa di 2 caratteri); i due campi sono separati da uno spazio.
Al fine di esempio, si vedano i file omonimi forniti nello StartKit.

Esercizio 1 – Struttura dati Distributore e funzioni di lettura/stampa (mod. element.h e dist.h/dist.c)

Si definisca un'opportuna struttura dati **`Distributore`**, al fine di rappresentare i dati relativi a ogni distributore. Si definisca la funzione:

```
Distributore * leggiDistributori(char* fileName, int *dim);
```

che, ricevuto in ingresso il nome di un file di testo rappresentante l'elenco dei distributori, restituisca un array di strutture dati di tipo **`Distributore`** allocato dinamicamente (della dimensione minima necessaria), contenente tutte le informazioni presenti nel file il cui nome è passato come parametro. Tramite il parametro **`dim`** la funzione deve restituire la dimensione del vettore allocato. La funzione abbia cura di gestire anche il caso in cui il file risulti essere eventualmente vuoto (restituisca cioè zero come dimensione del vettore: in tal caso non dovrà essere stata allocata dinamicamente alcuna memoria, ovviamente, e il puntatore restituito come risultato avrà valore **`NULL`**).
Si definisca la funzione:

```
void stampaDist(Distributore * v, int dim);
```

che, ricevuto in ingresso un array di strutture dati di tipo Distributore e la dimensione di tale array, stampi a video l'elenco dei distributori con tutte le informazioni relative. In particolare, poichè i prezzi sono memorizzati in centesimi di euro tramite un intero, si abbia cura di stampare i prezzi nella notazione in euro. Ad esempio, se la benzina costa 180 centesimi di euro, a video si stampi il prezzo "1.80".
Il candidato abbia cura di realizzare nel main opportuni test al fine di verificare il corretto funzionamento delle funzioni di cui sopra.

Fondamenti di Informatica T-1, 2011/2012 – Modulo 2

Prova d'Esame 2A di Giovedì 12 Gennaio 2012 – tempo a disposizione 2h

Esercizio 2 – Estrazione delle province e Ordinamento (moduli element.h/.c e dist.h/.c)

Il candidato definisca una procedura:

```
list estrai(char * fileName, char * regione);
```

che ricevuti come parametri il nome di un file contenente le coppie regione-provincia, e una stringa contenente il nome di una regione, restituisca una lista contenente come elementi le sigle delle province relative alla regione specificata come parametro. Tali informazioni sono disponibili nel file indicato come parametro. Al fine di riutilizzare agilmente le librerie sulle liste viste a lezione, si suggerisce di definire un nuovo tipo **`Provincia`** per rappresentare la sigla della provincia.
Il candidato definisca poi una procedura:

```
void ordina(Distributore * v, int dim);
```

tale che, ricevuti un vettore di strutture dati di tipo **`Distributore`** e la dimensione di tale vettore, ordini tale vettore in ordine lessicografico in base alla sigla della provincia. Si noti che purtroppo il file con l'elenco dei distributori non usa una notazione sempre coerente: infatti alcune volte le sigle delle province sono tutte con caratteri maiuscoli, mentre altre volte alcuni caratteri sono maiuscoli e altri minuscoli. La funzione di confronto tra le stringhe contenenti le province dovrà quindi tenere conto opportunamente di tale differenza: si vuole ottenere cioè un confronto "case insensitive". A tal scopo, il candidato realizzi una funzione di appoggio:

```
void diminuisci(char * dest, char * source);
```

che, ricevuto in ingresso una stringa ben formata **`source`**, copi in **`dest`** tutti i caratteri di **`source`** avendo cura di trasformare eventuali caratteri maiuscoli in minuscoli. Confrontando le stringhe ottenute non ci sarà più il problema delle maiuscole/minuscole.
Il candidato abbia cura di realizzare nel main opportuni test al fine di verificare il corretto funzionamento delle funzioni di cui sopra.

Esercizio 3 –Scelta del distributore più economico (modulo dist.h/dist.c)

Si definisca una funzione:

```
Distributore * economico(Distributore * v, int dimV, list elenco, int * dimR);
```

che, ricevuta come parametro un vettore di strutture dati di tipo **`Distributore`** e la sua dimensione **`dimV`**, e una lista di province passate tramite il parametro elenco, restituisca un nuovo vettore di strutture dati di tipo **`Distributore`** e la sua dimensione **`dimR`** (non necessariamente della dimensione minima). Nel vettore restituito come risultato dovranno essere inseriti, per ogni provincia specificata nella lista elenco, il distributore sito in tale provincia con il prezzo minore per la benzina. Qualora non esista un distributore per la provincia specificata, nel vettore restituito come risultato non dovrà essere inserita alcuna informazione relativamente a tale provincia. Il candidato rammenti anche il problema della notazione differente con maiuscole/minuscole nella specifica della sigla delle province.

Esercizio 4 – Stampa dei distributori più economici, e de-allocazione memoria (main.c)

Il candidato realizzi nella funzione **`main(…)`** un programma che chieda all'utente il nome di una regione (per semplicità, si supponga che tale regione sia presente nel file "regioni.txt"); il programma deve stampare a video, per ogni provincia della regione, il distributore più economico per la benzina verde.

Al termine del programma, il candidato abbia cura di de-allocare tutta la memoria allocata dinamicamente, ivi compresa la memoria allocata per le liste.

Fondamenti di Informatica T-1, 2011/2012 – Modulo 2

Prova d'Esame 2A di Giovedì 12 Gennaio 2012 – tempo a disposizione 2h

"element.h":

```
#include <string.h>

#ifndef _ELEMENT_H
#define      _ELEMENT_H

#define DIM_PROV 3
#define DIM_NOME 256

typedef struct {
       char prov[DIM_PROV];
       char nome[DIM_NOME];
       int verde;
       int gasolio;
} Distributore;


typedef struct {
       char prov[DIM_PROV];
} Provincia;

typedef Provincia element;

int compare(Distributore d1, Distributore d2);

#endif /* _ELEMENT_H */
```

"element.c":

```
#include "element.h"

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


int compare(Distributore d1, Distributore d2) {
       char temp1[DIM_PROV];
       char temp2[DIM_PROV];

       diminuisci(temp1, d1.prov);
       diminuisci(temp2, d2.prov);

       return strcmp(temp1, temp2);
}
```

Fondamenti di Informatica T-1, 2011/2012 – Modulo 2

Prova d'Esame 2A di Giovedì 12 Gennaio 2012 – tempo a disposizione 2h

"list.h"

```
#ifndef LIST_H
#define LIST_H

#include "element.h"

typedef struct      list_element
{
    element value;
    struct list_element  *next;
} item;

typedef  item* list;

typedef  int boolean;

/* PRIMITIVE  */
list emptylist(void);
boolean empty(list);
list cons(element, list);
element head(list);
list tail(list);

void showlist(list l);
void freelist(list l);
//int member(element el, list l);
//list insord_p(element el, list l);

#endif
```

"list.c":

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

#include "list.h"

/* OPERAZIONI PRIMITIVE */
list  emptylist(void)            /* costruttore lista vuota */
{
    return NULL;
}

boolean  empty(list l)     /* verifica se lista vuota */
{
    return (l==NULL);
}

list  cons(element e, list l)
{
  list t;       /* costruttore che aggiunge in testa alla lista */
  t=(list)malloc(sizeof(item));
  t->value=e;
  t->next=l;
  return(t);
}
```

Fondamenti di Informatica T-1, 2011/2012 – Modulo 2

Prova d'Esame 2A di Giovedì 12 Gennaio 2012 – tempo a disposizione 2h

```
element head(list l) /* selettore testa lista */
{
  if (empty(l)) exit(-2);
  else return (l->value);
}

list  tail(list l)         /* selettore coda lista */
{
  if (empty(l)) exit(-1);
  else return (l->next);
}


void showlist(list l) {
       element temp;
       if (!empty(l)) {
              temp = head(l);
              printf("%s ", temp.prov);
              return showlist(tail(l));
       }
       else {
              printf("\n");
              return;
       }
}


void freelist(list l) {
       if (empty(l))
              return;
       else {
              freelist(tail(l));
              free(l);
       }
       return;
}
```

"dist.h":

```
#ifndef _DIST_H
#define _DIST_H

#include "element.h"
#include "list.h"

Distributore * leggiDistributori(char* fileName, int *dim);
void stampaDist(Distributore * v, int dim);

list estrai(char * fileName, char * regione);
void diminuisci(char * dest, char * source);
void ordina(Distributore * v, int dim);

Distributore * economico(Distributore * v, int dimV, list elenco, int * dimR);

#endif
```

Fondamenti di Informatica T-1, 2011/2012 – Modulo 2

Prova d'Esame 2A di Giovedì 12 Gennaio 2012 – tempo a disposizione 2h

"dist.c":

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

#include "dist.h"
#include "list.h"


Distributore * leggiDistributori(char* fileName, int *dim) {
	Distributore * result;
	Distributore temp;
	FILE * fp;
	int i;
	int ok;

	*dim = 0;
	result = NULL;

	if ((fp=fopen(fileName, "rt")) == NULL) {
		printf("Errore durante l'apertura del file %s\n", fileName);
		system("pause");
		// exit(1); // non devo per forza sempre terminare...
	}
	else {
		// prima parte: conto quanti elementi ci sono nel file
		while (fscanf(fp, "%s%s%d%d", temp.prov, temp.nome, &temp.verde,
&temp.gasolio) == 4)
			*dim = *dim + 1;

		if (*dim > 0) {
			// alloco la memoria necessaria
			result = (Distributore *) malloc(*dim * sizeof(Distributore));

			// rileggo il file e copio le informazioni in memoria
			rewind(fp);
			ok = 1;
			i = 0;
			while (i<*dim && ok) {
				if (fscanf(fp, "%s%s%d%d", temp.prov, temp.nome, &temp.verde,
&temp.gasolio) == 4) {
					result[i] = temp;
					i++;
				}
				else
					ok = 0;
			}
			*dim = i;
		}

		fclose(fp);
	}
	return result;
}
```

Fondamenti di Informatica T-1, 2011/2012 – Modulo 2

Prova d'Esame 2A di Giovedì 12 Gennaio 2012 – tempo a disposizione 2h

```
void stampaDist(Distributore * v, int dim) {
	int i;

	for (i=0; i<dim; i++)
		printf("%s %s Benz.: %.2f, Gasolio: %.2f\n", v[i].prov, v[i].nome, ((float)
v[i].verde)/100, ((float) v[i].gasolio)/100);
	return;
}


list estrai(char * fileName, char * regione) {
	FILE * fp;
	list result;
	Provincia temp;
	char tempRegione[DIM_NOME];

	result = emptylist();
	if ((fp=fopen(fileName, "rt")) == NULL) {
		printf("Errore durante l'apertura del file %s\n", fileName);
		system("pause");
	}
	else {
		while (fscanf(fp, "%s%s", tempRegione, temp.prov) == 2)
			if (strcmp(tempRegione, regione) == 0)
				result = cons(temp, result);
		fclose(fp);
	}
	return result;
}


void diminuisci(char * dest, char * source) {
	while(*source != '\0') {
		if (*source > 'A' && *source < 'Z')
			*dest = *source - 'A' + 'a';
		else
			*dest = *source;
		source++;
		dest++;
	}
	*dest = '\0';
	return;
}


void scambia(Distributore *a, Distributore *b) {
	Distributore tmp = *a;
	*a = *b;
	*b = tmp;
}
// bubble sort
void ordina(Distributore * v, int n) {
	int i, ordinato = 0;
	while (n>1 && !ordinato) {
		ordinato = 1;
```

```
            for (i=0; i<n-1; i++)
                  if (compare(v[i],v[i+1])>0) {
                        scambia(&v[i],&v[i+1]);
                        ordinato = 0;
                  }
            n--;
      }
}

Distributore * economico(Distributore * v, int dimV, list elenco, int * dimR) {
      Distributore * result;
      Distributore record;
      int i;

      *dimR = 0;
      result = (Distributore *) malloc(dimV * sizeof(Distributore));
      while (!empty(elenco)) {
            record.verde = -1;
            //strcpy(record.prov, head(elenco).prov);
            strcpy(record.prov, elenco->value.prov);
            for (i=0; i<dimV; i++)
                  if (compare(record, v[i]) == 0)
                        if (v[i].verde < record.verde || record.verde == -1)
                              record = v[i];

            if (record.verde > -1) {
                  result[*dimR] = record;
                  *dimR = *dimR + 1;
            }
            elenco = tail(elenco);
      }
      return result;
}
```

Fondamenti di Informatica T-1, 2011/2012 – Modulo 2

Prova d'Esame 2A di Giovedì 12 Gennaio 2012 – tempo a disposizione 2h

"main.c":

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

#include "element.h"
#include "dist.h"

int main(void) {

	// test es 1
	{
		int dim;
		Distributore *res = leggiDistributori("distributori.txt", &dim);
		stampaDist(res, dim);
		printf("\n\n");
		free(res);
	}

	// test es 2
	{
		int dim;
		Distributore *res;
		list elenco;
		elenco = estrai("regioni.txt", "Emilia_Romagna");
		showlist(elenco);

		res = leggiDistributori("distributori.txt", &dim);
		ordina(res, dim);
		stampaDist(res, dim);
		printf("\n\n");

		free(res);
		freelist(elenco);
	}

	// test es 3 e 4
	{
		int dim;
		int dimEco;
		Distributore *res;
		Distributore * eco;
		list elenco;
		elenco = estrai("regioni.txt", "Emilia_Romagna");
		res = leggiDistributori("distributori.txt", &dim);
		eco = economico(res, dim, elenco, &dimEco);
		stampaDist(eco, dimEco);
		printf("\n\n");

		free(res);
		free(eco);
		freelist(elenco);
	}
	return 0;
}
```

Fondamenti di Informatica T-1, 2011/2012 – Modulo 2

Prova d'Esame 2A di Giovedì 12 Gennaio 2012 – tempo a disposizione 2h

"distributori.txt":

```
RN Benzina_al_mare 185 180
BO Da_mario_carburanti 175 170
BO Alessandro_benzine 174 168
Bo Federico_gas_gas 170 160
Mo Da_MARIO_carburanti 178 170
FC Da_Mario_Benzina_e_Piadina 160 140
MO Col_pieno_sei_felice 160 153
```

"regioni.txt":

```
Emilia_Romagna BO
Emilia_Romagna FE
Emilia_Romagna FC
Emilia_Romagna MO
Emilia_Romagna PR
Emilia_Romagna PC
Emilia_Romagna RA
Emilia_Romagna RE
Emilia_Romagna RN
Toscana AR
Toscana FI
Toscana GR
Toscana LI
Toscana LU
Toscana MS
Toscana PI
Toscana PT
Toscana PO
Toscana SI
```