Fondamenti di Informatica L-A (A.A. 2004/2005) - Ingegneria Informatica
Prof.ssa Mello & Prof. Bellavista – Prova d'Esame di Lunedì 20 Dicembre 2004 – durata 2h30m
COMPITO B

ESERCIZIO 1 (14 punti)

Una filiale di banca, sita in via Rizzoli, registra i depositi di denaro effettuati presso la sua filiale su un file binario di nome "**`rizzoli.dat`**". In particolare ogni registrazione in tale file è una struttura contenente i seguenti campi (il candidato provveda a definire opportunamente tale struttura):

- Codice del conto corrente (una stringa, di al più 15 caratteri, senza spazi);
- giorno del deposito (un intero, rappresentante un unico giorno nell'ambito del corrente anno);
- ammontare di denaro depositato (un float).

Si vuole realizzare un programma che selezioni i depositi effettuati in base al giorno in cui sono stati fatti e visualizzi alcune informazioni di tipo riassuntivo. Con maggior dettaglio:

1. Il candidato scriva una funzione **`readDeposit(…)`** che legga tutte le strutture dal file "**`rizzoli.dat`**". Il numero di depositi registrati sul file non è noto a priori, quindi il candidato provveda a leggere una prima volta il file (per stabilire quanti record sono registrati), quindi allochi memoria a sufficienza e poi rilegga il file da principio per memorizzare i dati (si usi a tal scopo la procedura **`void rewind(FILE * f)`**, che riporta la testina di lettura a inizio file). La funzione **`readDeposit(…)`** deve avere come parametri un puntatore al file da leggere (tale file quindi deve essere già stato aperto opportunamente per consentirne la lettura) e un intero passato per riferimento, tramite il quale verrà salvato il numero di elementi letti. La funzione deve restituire un puntatore ad una opportuna struttura dati definita dall'utente in cui sono stati salvati i dati letti dal file. **(punti 7)**
2. Il candidato realizzi poi un programma C che, utilizzando la funzione **`readDeposit(…)`**, legga tutte le informazioni dal file binario "**`rizzoli.dat`**": a tal scopo si provveda ad aprire il file, prima di invocare la funzione. Il programma poi chieda all'utente di inserire il primo e l'ultimo giorno dell'intervallo di interesse, e stampi a video tutte le informazioni sui depositi effettuati in tale intervallo. Il programma provveda poi a calcolare e stampare a video il solo valore massimo dei depositi effettuati in tale intervallo temporale; si consideri che tutti i valori di deposito sono (ovviamente) positivi. **(punti 7)**

ESERCIZIO 2 (6 punti)

Si scriva una funzione **`merge_diff()`** che date in ingresso due liste **`a`** e **`b`** di interi, ordinati in senso crescente, restituisca una terza lista contenente gli interi non presenti in entrambe le liste, ancora ordinati in senso crescente. Le liste **`a`** e **`b`** possono avere lunghezze diverse. Ad esempio, con **`a=[1,3,4,6]`** e **`b=[2,3,4,7,12]`**, il risultato deve essere la lista **`[1,2,6,7,12]`**.
La funzione **`merge_diff()`** può essere realizzata in modo ricorsivo o iterativo, utilizzando il tipo di dato astratto **`list`**. Si possono utilizzare le sole operazioni primitive definite durante il corso (che quindi possono NON essere riportate nella soluzione). Non si possono usare altre funzioni di alto livello.

ESERCIZIO 3 (6 punti)

Il seguente programma C compila correttamente? In caso affermativo, quali sono i valori stampati a tempo di esecuzione (si motivi opportunamente la risposta data)?

```
#include <stdio.h>
#include <stdlib.h>
#define MAX 16
int dim = MAX - 9;

void calcola(char * t1, char * t2, int * m) {
      int i=0;
      for (i=0; (t2[i] != '\0'); i++);

      for (; (i >=0); i=i-1) {
            t1[(*m)] = *(t2+i);
            *m = *m-1;
      }
}

int main () {
      char * s;
      int i;

      s = (char *) malloc(MAX * sizeof(char));
      calcola(s, "Ferrara", &dim);

      for (i=0; (s[i] != '\0') && (s[i+1] != '\0'); i=i+2)
            printf("%c", s[i]);

      printf("\ndim vale adesso: %d\n",dim);
      return 0;
}
```

ESERCIZIO 4 (4 punti)

Si consideri la seguente funzione `int procedura(double a, double b)`:

```
int procedura(double a, double b){
      a = a-b;
      --b;
      if (a > 0) return procedura(a, b);
            else return b;
}
```

Si scriva il risultato della funzione quando invocata come `procedura(8.0, 3.5)` e si disegnino i corrispondenti record di attivazione (si faccia particolare attenzione al fatto che la funzione restituisce un valore intero). La funzione è ricorsiva tail?

ESERCIZIO 5 (2 punti)

Un elaboratore rappresenta i numeri interi su 8 bit dei quali 7 sono dedicati alla rappresentazione del modulo del numero e uno al suo segno. Indicare come viene svolta la seguente operazione aritmetica e determinarne il risultato traslandolo poi in decimale per la verifica:

108 - 23

SOLUZIONE

ESERCIZIO 1

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

#define DIM 16

typedef struct {
      char code[DIM];
      int day;
      float amount;
} deposit;

deposit * readDeposit(FILE * f, int * size) {
      deposit temp;
      deposit * result;

      // conta quanti elementi ci sono...
      for (*size = 0; fread(&temp, sizeof(deposit), 1, f) > 0; (*size)++);
      result = malloc( sizeof(deposit) * *size);

      // legge tutti gli elementi
      rewind(f);
      for (*size = 0; fread((result + *size), sizeof(deposit), 1, f) > 0;
            (*size)++);

      return result;
}

int main() {
      deposit * records;
      int size = 0, i;
      int firstDay, lastDay;
      float max = -1;
      FILE * f;

      if ( (f = fopen("rizzoli.dat", "r")) == NULL ) {
            printf ("Error opening the file %s\n", "rizzoli.dat");
            exit(-1);
      }
      records = readDeposit(f, &size);
      fclose(f);  // chiude il file

      printf("Insert first & last day: ");
      scanf("%d%d", &firstDay, &lastDay);


      for (i=0; i< size; i++)
            if ( (records[i].day > firstDay) && (records[i].day < lastDay)) {
                  printf("%s %d %6.2f\n", records[i].code, records[i].day,
records[i].amount);
                  if (records[i].amount > max)  max = records[i].amount;
      }
      printf("\nMaximum deposit in the selected period: %6.2f\n\n", max);


      free(records);
      return 0;
}
```

ESERCIZIO 2

```
list merge_diff(list a, list b) {
      if (empty(a)) return b;
      else if (empty(b)) return a;
            else if (head(a) == head(b)) return merge_diff(tail(a), tail(b));
                  else if (head(a) < head(b))
                              return cons(head(a), merge_diff(tail(a), b));
                        else return cons(head(b), merge_diff(a, tail(b)));
}
```

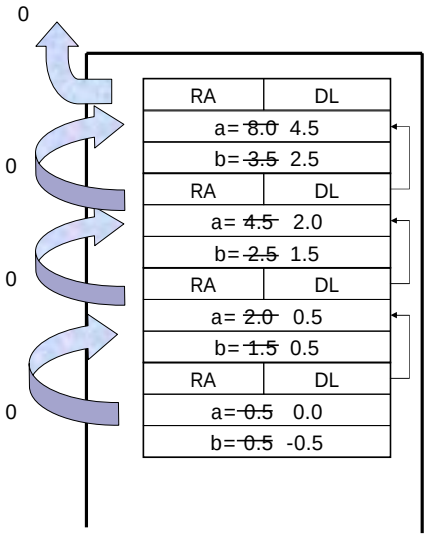
ESERCIZIO 3

Il programma è corretto sintatticamente e stampa:
Fra
dim adesso vale: -1

Il programma main, come prima cosa, alloca spazio in memoria per 16 (il valore di MAX) caratteri; tale spazio è puntato dalla variabile s. Quindi invoca la funzione calcola(), con parametri s, "Ferrara", e l'indirizzo dell'intero dim (il cui valore è 7). La funzione calcola() determina in un primo ciclo la lunghezza della stringa passata in ingresso, tramite controllo su presenza del terminatore di stringa. La variabile i, che rappresenta appunto la lunghezza di t2, viene a valere 7. Nel secondo ciclo si provvede a copiare t2 in t1, terminatore compreso, partendo però dall'ultimo carattere; si noti che ad ogni ciclo vengono decrementati opportunamente sia i che m. Al termine della funzione calcola(), la stringa "Ferrara" è stata copiata nello spazio puntato da s, e la variabile globale dim vale -1. Infine, nel main è presente un ultimo ciclo, che provvede a stampare il contenuto della memoria indirizzata da s. A tal scopo si usa un indice i (locale al main). La variabile i viene però incrementata di 2 ad ogni iterazione, e quindi vengono stampati solo i caratteri di indice pari. Viene stampato anche un terminatore, e poi il valore di dim, che per quanto detto prima vale ancora -1.

ESERCIZIO 4

La funzione è ricorsiva tail e restituisce il valore 0, attraverso la seguente sequenza di record di attivazione:



ESERCIZIO 5

```
108->       0   1101100
23->        0   0010111
```

Tra i numeri si esegue una sottrazione ottenendo: `0  1010101`
che vale 85 in base dieci.