

**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 A**

**ESERCIZIO 1 (14 punti)**

Una clinica privata registra la propria contabilità annuale su un file binario di nome “**pazienti.dat**”. In particolare ogni registrazione in tale file è una struttura contenente i seguenti campi (il candidato provveda a definire tale struttura opportunamente nel compito):

- nome del paziente (al più 63 caratteri, senza spazi)
- giorno di uscita dalla clinica (un intero, rappresentante un unico giorno nell’ambito del corrente anno);
- importo fatturato (un float);

Si vuole realizzare un programma che selezioni i pazienti in base al giorno di uscita e visualizzi alcune informazioni di tipo economico. Con maggior dettaglio:

1. Il candidato scriva una funzione **readPatients(...)** che legga tutte le strutture dal file “**pazienti.dat**”. Il numero di pazienti 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 **readPatients(...)** deve avere come parametri un array di caratteri rappresentante il nome del file 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 **readPatients(...)**, legga tutte le informazioni dal file binario “**pazienti.dat**”; il programma poi chieda all’utente di inserire un giorno iniziale e un giorno finale, e stampi a video tutte le informazioni sui pazienti che sono usciti dalla clinica nell’intervallo compreso tra quei due giorni. Il programma provveda poi a calcolare il valore medio degli importi fatturati a tali utenti. **(punti 7)**

**ESERCIZIO 2 (6 punti)**

Si scriva una funzione **merge()** che date in ingresso due liste **a** e **b** di interi, ordinati in senso crescente, restituisca una terza lista contenente gli interi di entrambe le liste, ancora ordinati in senso crescente, da cui siano eliminati eventuali elementi doppi. Le liste **a** e **b** possono avere lunghezze diverse. Ad esempio, con **a=[1,3,4]** e **b=[2,3,4,7,12]**, il risultato deve essere la lista **[1,2,3,4,7,12]**.

La funzione **merge()** 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 10

int dim = MAX - 3;

void esegui( char * s1, char * s2, int * m) {
    int i=0;
    for (i=0; (s2[i] != '\0'); i++);

    for (; (*m >= 0); (*m)--) {
        *(s1+ *m) = s2[i];
        i--;
    }
}

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

    t = (char *) malloc(MAX * sizeof(char));
    esegui(t, "Bologna", &dim);

    for (i=dim+1; (t[i] != '\0'); dim++, i=dim+1)
        printf("%c", t[i]);

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

### **ESERCIZIO 4 (4 punti)**

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

```
int funzione(float a, float b) {
    a = a+b;
    b--;
    if (b > 0) return funzione(a, b);
    else return a; }
```

Si scriva il risultato della funzione quando invocata come `funzione(2.5, 3.5)` e si disegnano 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:

**39 - 94**

## SOLUZIONE

### ESERCIZIO 1

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

#define DIM 64

typedef struct {
    char name[DIM];
    int day;
    float bill;
} ricovero;

ricovero * readPatients(char * fileName, int * size) {
    int i;
    FILE * f;
    ricovero temp;
    ricovero * result;

    if ( (f = fopen(fileName, "r")) == NULL ) {
        printf ("Error opening the file %s\n", fileName);
        exit(-1);
    }

    for (*size = 0; fread(&temp, sizeof(ricovero), 1, f) > 0; (*size)++);
    result = malloc(sizeof(ricovero) * (*size));

    rewind(f);
    for (*size = 0; fread((result + *size), sizeof(ricovero), 1, f) > 0;
        (*size)++);

    fclose(f);
    return result;
}

int main() {
    ricovero * patients;
    int size = 0, i, startDay, endDay;
    int selected = 0;
    float total = 0;

    patients = readPatients("pazienti.dat", &size);

    printf("Insert start & end day: ");
    scanf("%d%d", &startDay, &endDay);

    for (i=0; i<size; i++)
        if ( (patients[i].day > startDay) && (patients[i].day < endDay)) {
            printf("%s %d %6.2f\n", patients[i].name, patients[i].day,
                patients[i].bill);
            total = total + patients[i].bill;
            selected++;
        }
    printf("\nAverage bill in the selected period: %6.2f\n\n", total/selected);

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

## ESERCIZIO 2

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

## ESERCIZIO 3

Il programma è corretto sintatticamente e stampa:

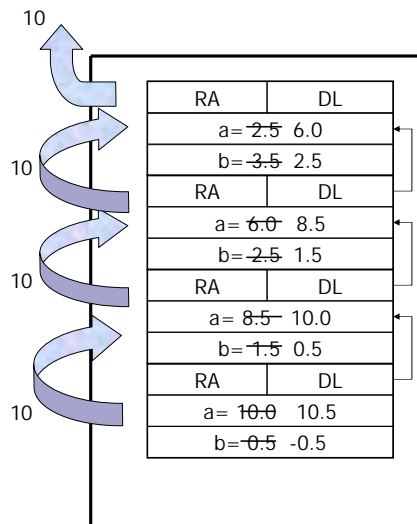
**Bologna**

**dim adesso vale: 6**

Il programma main, come prima cosa, alloca spazio in memoria per 10 (il valore di MAX) caratteri; tale spazio è puntato dalla variabile t. Quindi invoca la funzione esegui(), con parametri t, "bologna", e l'indirizzo dell'intero dim (il cui valore è 7). La funzione esegui() 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 s2, viene a valere 7. Nel secondo ciclo si provvede a copiare s2 in s1, terminatore compreso, partendo però dall'ultimo carattere; si noti che ad ogni ciclo vengono decrementati opportunamente sia i che m. Al termine della funzione esegui(), la stringa "Bologna" è stata copiata nello spazio puntato da t, 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 t. A tal scopo si usa un indice i (locale al main) opportunamente inizializzato in modo da stampare la corretta zona di memoria. Viene stampato anche un terminatore, e poi il valore di dim, che a causa dei vari incrementi alla fine vale 6.

## ESERCIZIO 4

La funzione è ricorsiva tail e restituisce il valore 10, attraverso la seguente sequenza di record di attivazione:



## ESERCIZIO 5

39-> 0 0100111

94-> 0 1011110

Tra i moduli dei numeri si esegue una sottrazione ottenendo:

1 0110111

che vale -55 in base dieci.