

Esame di Fondamenti di Informatica Ingegneria Meccanica (A–O) Appello del 22/2/2002

Esercizio 1

Un elaboratore adotta per i **numeri interi** una rappresentazione in complemento a due su un byte e per i **numeri reali** una rappresentazione in virgola mobile con un byte per la mantissa normalizzata in segno/modulo (si usi il primo bit della mantissa per rappresentare il segno) ed un byte per l'esponente in complemento a due.

Si consideri l'espressione:

$$-43 - 66 + 86.75$$

Indicare il risultato dell'espressione ottenuto eseguendo il calcolo con l'elaboratore dato. Mostrare i passaggi intermedi eseguiti dall'elaboratore con particolare riferimento alle operazioni ed alla rappresentazione interna in binario, nonché i relativi eventuali errori compiuti. Scrivere il risultato finale in codifica decimale e confrontarlo con il risultato atteso.

Esercizio 2

Sia dato il seguente programma C:

```
#include <stdio.h>
#define n 4;

int F(int V[], int i) {
    if (i>0) return V[i+1]+F(V, --i);
    else return V[i];
}

main() {
    int V[n];
    int i;

    for(i=0; i<n; i++) V[i]=i;
    printf("%d\n", F(V, n-1));
    for(i=0; i<n; i++) printf("%d\t", V[i]);
    printf("\n");
}
```

Scrivere l'output prodotto dal programma ed illustrare la dinamica dei record di attivazione sullo stack. Motivare opportunamente le risposte.

Esercizio 3

Al festival della canzone di Santa Rema partecipano, ogni anno, diversi autori che presentano una canzone inedita. Il pubblico del festival di Santa Rema può votare per la canzone che reputa migliore. Vincitore del festival risulta l'autore la cui canzone riceve il maggior numero di voti.

Nell'edizione di quest'anno, per automatizzare la procedura di voto, le informazioni sulle canzoni sono memorizzate all'interno di un file binario, chiamato "CANZONI.DAT", in cui, per ogni record, viene memorizzato il nome del cantante, il titolo della canzone ed il numero di voti ricevuti.

La giuria del festival riceve i dati relativi ai voti del pubblico all'interno di un file di testo, chiamato "VOTI.TXT", che in ogni linea contiene il nome della canzone votata.

Si scriva un programma C per la giuria del festival di Santa Rema che:

1. Legga il contenuto del file "CANZONI.DAT" e lo memorizzi all'interno di un vettore V. Il massimo numero di canzoni presenti al festival è 50.
2. Legga dal file "VOTI.TXT" i voti espressi dal pubblico e provveda ad aggiornare in maniera opportuna il vettore V. A tal scopo, si supponga l'esistenza di una procedura *aggiorna* che, dati un vettore di canzoni ed una stringa rappresentante il nome di una canzone, provveda ad incrementare di 1 il numero di voti ricevuti da tale canzone. Si supponga che non esistano canzoni aventi lo stesso nome.
3. Dati, da tastiera, un intero rappresentante l'indice di una canzone nel vettore ed un secondo intero che rappresenta un numero di voti (eventualmente negativo), provveda ad aggiornare il numero di voti relativi alla canzone in questione (la giuria può così aumentare o diminuire arbitrariamente il numero di voti di una qualsiasi canzone).
4. Stampi a video il titolo della canzone (ed il relativo nome del cantante) che ha ricevuto il maggior numero di voti.
5. Si scriva il codice della procedura *aggiorna* (a tal scopo si utilizzi la funzione `strcmp`, dichiarata all'interno del file "string.h", che, date due stringhe, restituisce 0 se e solo se le due stringhe sono identiche).

Esercizio 4

Discutere il concetto di campo di visibilità delle variabili in C.

Soluzione Esercizio 1

Rappresentazione – **43**:

$$(43)_{10} = (101011)_2$$

rappr. interna = **1101 0101**

Rappresentazione – **66**:

$$(66)_{10} = (1000010)_2$$

rappr. interna = **1011 1110**

Risultato – **43 – 66**:

$$1101\ 0101 +$$

$$\underline{1011\ 1110} =$$

$$1001\ 0011 = (-109)_{10}$$

Risultato – **109 + 86.75**:

Normalizzazione del risultato: -0.11011010×2^7

Nessun incolonnamento

$$0.1101\ 1010 \times 2^7 -$$

$$\underline{0.1010\ 1101} \times 2^7 =$$

$$0.0010\ 1101 \times 2^7$$

Normalizzando: -0.101101×2^5

Errore di cancellazione!

byte esponente = **0000 0101**

byte mantissa = **1011 0100**

risultato: $(-10110.1)_2 = (-22.5)_{10}$

risultato atteso: $(-22.25)_{10}$

Soluzione Esercizio 2

Il risultato del programma è il seguente:

```
6
0  1  2  3
```

Infatti, ogni invocazione della funzione `f` produce, come risultato, la somma dell'*i*-esimo elemento del vettore `V` e del valore restituito dalla funzione `f` chiamata con un valore di *i* decrementato di 1. La ricorsione ha termine quando *i* assume il valore 0. Di conseguenza, l'effetto della funzione è quello di sommare i valori contenuti nel vettore. Il valore restituito dalla funzione è pertanto 6, mentre i valori del vettore sono immutati.

Soluzione Esercizio 3

```
#include <stdio.h>
#include <string.h>
#define DIM 50

typedef struct {
    char cantante[50];
    char titolo[30];
    int voti;
} canzone;

void aggiorna(canzone V[], char titolo[]);

main() {
    canzone V[DIM];
    FILE *fp=fopen("CANZONI.DAT", "rb");
    int n, min=0, imin, i, voti;

    n=fread(V, sizeof(canzone), DIM, fp);
    fclose(fp); /* fine domanda 1. */
    fp=fopen("VOTI.TXT", "rt");
    while(!feof(fp)) {
        char s[30];

        fscanf(fp, "%s", s);
        aggiorna(V, s);
    } /* fine domanda 2. */
    scanf("%d %d", &i, &voti);
    V[i].voti+=voti; /* fine domanda 3. */
    for(i=0; i<n; i++)
        if(V[i].voti>min) {
            min=V[i].voti;
            imin=i;
        }
    printf("%s %s\n", V[imin].cantante, V[imin].titolo);
    /* fine domanda 4. */
}

void aggiorna(canzone V[], char titolo[]) {
    int i;

    for(i=0; ; i++)
        if(!strcmp(V[i].titolo, titolo)) {
            V[i].voti++;
            return;
        }
} /* fine domanda 5. */
```