

Esame di Fondamenti di Informatica Ingegneria Meccanica (A–O) Appello del 17/6/2002

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:

$$(115 - 38)/16 - 14.63$$

Indicare il risultato dell'espressione ottenuto eseguendo il calcolo con l'elaboratore dato (non si esegua la divisione intera). 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 *p, int i) {
    if(++i<n) {
        *(p+1)+=*p;
        return F(++p, i);
    }
    return *p;
}

main() {
    int j, V[n]={1, 2, 3, 4};

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

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

Esercizio 3

La federazione calcistica dello stato caraibico di St. Marquez sta preparando la squadra nazionale per le qualificazioni ai prossimi mondiali. Il CT Lope Amaral sceglie i giocatori da schierare per ogni partita in base alle statistiche ottenute da ciascun atleta nelle partite del campionato domestico. Tali statistiche sono contenute all'interno di un file di testo chiamato "PLAYERS.TXT". In ogni riga del file sono contenuti il nome ed il cognome del giocatore, il ruolo (un semplice carattere: G = portiere, D = difensore, M = centrocampista, A = attaccante), le partite disputate, i minuti giocati e i gol segnati.

Si scriva un programma C per il selezionatore della nazionale di St. Marquez che:

1. Legga il contenuto del file "PLAYERS.TXT" e lo memorizzi all'interno di un vettore V. Si supponga che il massimo numero di giocatori sia 200.
2. Legga da tastiera un carattere corrispondente ad un ruolo (secondo la convenzione precedentemente indicata) e provveda ad inserire in un secondo vettore V1 le informazioni sui soli giocatori aventi tale ruolo. A tal scopo, si supponga l'esistenza di una procedura `inserisci` che, dato un vettore di giocatori, un intero rappresentante una posizione all'interno del vettore e le informazioni su un giocatore, provveda a copiare le informazioni su tale giocatore nella posizione specificata nel vettore.
3. Provveda a stampare le informazioni sull'attaccante più prolifico (ovvero quello avente la media gol segnati/minuti giocati più alta).
4. Si scriva il codice della procedura `inserisci` (a tal scopo si utilizzi la funzione `strcpy`, dichiarata all'interno del file "string.h", che, date due stringhe, copia la seconda nella prima).

Esercizio 4

Illustrare l'organizzazione del file system MS-DOS.

Soluzione Esercizio 1

Rappresentazione **115**:

$$(115)_{10} = (1110011)_2$$

rappr. interna = **0111 0011**

Rappresentazione – **38**:

$$(38)_{10} = (100110)_2$$

rappr. interna = **1101 1010**

Rappresentazione **16**:

$$(16)_{10} = (10000)_2$$

rappr. interna = **0001 0000**

Risultato **115 – 38**:

$$0111\ 0011 +$$

$$\underline{1101\ 1010} =$$

$$0100\ 1101 = (77)_{10}$$

Risultato **77 / 16**: $1001101 / 2^4 = 100.1101 = (4.8125)_{10}$

Normalizzando: 0.10011010×2^3

Risultato **4.8125 – 14.63**:

Normalizzazione del risultato: -0.01001101×2^4

Nessun errore di incolonnamento

$$0.1110\ 1010 \times 2^4 -$$

$$\underline{0.0100\ 1101} \times 2^4 =$$

$$0.1001\ 1101 \times 2^4$$

Normalizzando: -0.10011101×2^4

Nessun errore di cancellazione!

byte esponente = **0000 0100**

byte mantissa = **1001 1101**

risultato: $(-1001.1101)_2 = (-9.8125)_{10}$

risultato atteso: $(-9.8175)_{10}$

Soluzione Esercizio 2

Il risultato del programma è il seguente:

10

1 3 6 10

Infatti, ad ogni invocazione della funzione F, l'elemento corrente del vettore viene sommato all'elemento precedente; quindi, la funzione F viene chiamata ricorsivamente incrementando il puntatore p all'elemento successivo. La ricorsione ha termine quando i assume il valore 3. Di conseguenza, l'effetto della funzione è quello di sommare i valori contenuti nel vettore. Il valore restituito dalla funzione è pertanto 10, mentre i valori del vettore sono variati come indicato.

Rappresentazione **14.63**:

$$(14.63)_{10} = (1110.1010)_2$$

Errore di troncamento!

Normalizzando: 0.11101010×2^4

byte esponente = **0000 0100**

byte mantissa = **0110 1010**

Soluzione Esercizio 3

```
#include <stdio.h>
#include <string.h>
#define DIM 200

typedef struct {
    char nome[30], cognome[10], ruolo;
    int partite, minuti, gol;
} giocatore;

void inserisci(giocatore V[], int i, giocatore e);

main() {
    giocatore V[DIM], V1[DIM];
    FILE *fp=fopen("PLAYER.TXT", "rt");
    int i, n=0, nl=0, imax;
    char c;
    float max=0;

    while(!feof(fp)) {
        fscanf(fp, "%s%s%c%d%d", V[n].nome, V[n].cognome,
            &V[n].ruolo, &V[n].partite, &V[n].minuti, &V[n].gol);
        n++;
    }
    fclose(fp); /* fine domanda 1. */
    scanf("%c", &c);
    for(i=0; i<n; i++)
        if(V[i].ruolo==c)
            inserisci(V1, nl++, V[i]);
    /* fine domanda 2. */
    for(i=0; i<n; i++) {
        if((V[i].ruolo=='a')&&(((float)V[i].gol)/V[i].minuti)>max) {
            imax=i;
            max=((float)V[i].gol)/V[i].minuti;
        }
    }
    printf("%s%s%c%d%d", V[imax].nome, V[imax].cognome,
        V[imax].ruolo, V[imax].partite, V[imax].minuti, V[imax].gol);
    /* fine domanda 3. */
}

void inserisci(giocatore V[], int i, giocatore e) {
    strcpy(V[i].nome, e.nome);
    strcpy(V[i].cognome, e.cognome);
    V[i].ruolo=e.ruolo;
    V[i].partite=e.partite;
    V[i].minuti=e.minuti;
    V[i].gol=e.gol;
} /* fine domanda 4. */
```