

# Esame di Fondamenti di Informatica Ingegneria Meccanica (A–O) Appello del 19/10/2001

## Compito A

### 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:

$$55.375 - 19 * 6$$

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 7

int V[N] = {1, 2, 3, 4, 5, 6, 7};

int F(int V[]) {
    return V[1]/V[0];
}

main() {
    int i;

    for(i=0; (i+1)<N; i+=2) V[4+i/2]=F(V+i);
    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

La segreteria della modernissima Università "La Sorbolona" deve preparare la documentazione relativa alle sessioni degli esami di laurea. A questo scopo, le informazioni sugli studenti immatricolati vengono contenute in un file binario, chiamato "STUDENTI.DAT". Per ogni studente vengono mantenute le informazioni riguardanti il nome, il cognome ed il numero di matricola. Il numero totale di studenti contenuti nel file è presente all'inizio del file stesso come numero intero.

I voti di tutti gli studenti sono contenuti in un unico file binario, chiamato "VOTI.DAT". Ciascun record di tale file contiene il numero di matricola dello studente, il codice del corso (un semplice intero) e la votazione ottenuta nell'esame.

Il nome ed il cognome dei candidati agli esami di laurea sono contenuti in un file di testo, chiamato "CANDIDATI.TXT".

Si scriva un programma C per la segreteria dell'Università "La Sorbolona" che:

1. A partire dal file "STUDENTI.DAT", crei un vettore V di dimensioni opportune contenente le informazioni su tutti gli studenti.
2. Utilizzando il vettore V, provveda a scrivere, su un file di testo chiamato "LAUREANDI.TXT" le informazioni (nome, cognome, matricola e media voti) relative ai soli studenti laureandi (ovvero agli studenti il cui nome e cognome sono contenuti nel file "CANDIDATI.TXT"). A tal scopo, si supponga l'esistenza di una funzione `media` che, dato il numero di matricola di uno studente, acceda al file "VOTI.DAT" e restituisca la media voti dello studente. Per il confronto tra stringhe 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.
3. Letto un valore da tastiera, stampi su video il nome ed il cognome di tutti i laureandi aventi una media non inferiore a tale valore.
4. Si scriva il codice della funzione `media`. Le informazioni sui voti non devono essere memorizzate in strutture interne, ma vanno lette volta per volta dal file "VOTI.DAT".

### Esercizio 4

Si illustri il ciclo di esecuzione di un'istruzione all'interno della CPU.

### Soluzione Esercizio 1

Rappresentazione 19:

$$(19)_{10} = (10011)_2$$

rappr. interna = **0001 0011**

Rappresentazione 6:

$$(6)_{10} = (110)_2$$

rappr. interna = **0000 0110**

Risultato 19 \* 5:

$$\begin{array}{r} 10011 \quad \times \\ \underline{110} \quad = \\ 100110 \\ 1001100 \\ \hline 1110010 \end{array}$$

Risultato = 0111 0010 = (114)<sub>10</sub>

Risultato 55.375 - 114:

Normalizzazione del risultato: 0.1110010 × 2<sup>7</sup>

Incolonnamento del numero più piccolo: 0.01101110 × 2<sup>7</sup>

**Errore di incolonnamento!**

$$0.1110 0100 \times 2^7 -$$

$$\underline{0.0110 1110} \times 2^7 =$$

$$0.0111 0110 \times 2^7$$

Normalizzando: - 0.11101100 × 2<sup>6</sup>

**Errore di cancellazione!**

byte esponente = **0000 0110**

byte mantissa = **1110 1100**

risultato: (- 111011)<sub>2</sub> = (- 59)<sub>10</sub>

risultato atteso: (- 58.625)<sub>10</sub>

### Soluzione Esercizio 2

Il risultato del programma è il seguente:

**1 2 3 4 2 1 0**

Infatti, alla prima invocazione della funzione F (i=0) viene calcolata la divisione intera tra i primi due elementi del vettore V (2/1 = 2) ed il risultato viene posto nel 5° elemento del vettore (4+0/2 = 4); alla seconda invocazione della funzione F (i=2) viene calcolata la divisione intera tra il quarto ed il terzo elemento del vettore V (4/3 = 1) ed il risultato viene posto nel 6° elemento del vettore (4+2/2 = 5); alla terza invocazione della funzione F (i=4) viene calcolata la divisione intera tra il sesto ed il quinto elemento del vettore V (1/2 = 0) ed il risultato viene posto nel 7° elemento del vettore (4+4/2 = 6). I valori degli elementi del vettore sono quindi quelli indicati.

Rappresentazione 55.375:

$$(55.375)_{10} = (110111.011)_2$$

**Errore di troncamento!**

Normalizzando: 0.11011101 × 2<sup>6</sup>

byte esponente = **0000 0110**

byte mantissa = **0101 1101**

### Soluzione Esercizio 3

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

typedef struct { char nome[20], cognome[30]; int matricola; } studente;
typedef struct { int matricola, codice, voto; } esame;

float media(int matricola);

main() {
    FILE *fp=fopen("STUDENTI.DAT", "rb"), *fpl, *fpc;
    int N, i;
    float valore;
    studente *V;
    fread(&N, sizeof(int), 1, fp);
    V=(studente *)malloc(N*sizeof(studente));
    fread(V, sizeof(studente), N, fp);
    fclose(fp); /* fine domanda 1 */
    fpl=fopen("LAUREANDI.TXT", "wt");
    fpc=fopen("CANDIDATI.TXT", "rt");
    while(!feof(fpc)) {
        char nome[20], cognome[30];
        fscanf(fpc, "%s %s", nome, cognome);
        for(i=0; i++)
            if(!strcmp(V[i].nome, nome) && !strcmp(V[i].cognome, cognome)) {
                fprintf(fpl, "%s %s %d %f\n", nome, cognome, V[i].matricola,
                    media(V[i].matricola));
                break;
            }
    }
    fclose(fpc);
    fclose(fpl); /* fine domanda 2 */
    scanf("%f", &valore);
    fp=fopen("LAUREANDI.TXT", "rt");
    while(!feof(fp)) {
        char nome[20], cognome[30];
        int matricola;
        float media;
        fscanf(fp, "%s %s %d %f\n", nome, cognome, &matricola, &media);
        if(media>valore) printf("%s %s\n", nome, cognome);
    }
    fclose(fp); /* fine domanda 3 */
    free(V);
}

float media(int matricola) {
    FILE *fp=fopen("VOTI.DAT", "rb");
    int N=0;
    float media=0;
    esame e;
    while(!feof(fp)) {
        fread(&e, sizeof(esame), 1, fp);
        if(e.matricola==matricola) {
            media+=e.voto;
            N++;
        }
    }
    return media/N;
}
```

# Esame di Fondamenti di Informatica Ingegneria Meccanica (A–O) Appello del 19/10/2001

## Compito B

### 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:

$$43.625 - 21 * 5$$

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 7

int V[N] = {7, 6, 5, 4, 3, 2, 1};

int F(int V[]) {
    return V[0]/V[1];
}

main() {
    int i;

    for(i=N; (i/2)>0; i-=2) V[i/2-1]=F(V+i-2);
    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

La segreteria della modernissima Università "La Sorbolona" deve preparare la documentazione relativa alle sessioni degli esami di laurea. A questo scopo, le informazioni sugli studenti immatricolati vengono contenute in un file binario, chiamato "STUDENTI.DAT". Per ogni studente vengono mantenute le informazioni riguardanti il nome, il cognome ed il numero di matricola.

I voti di tutti gli studenti sono contenuti in un unico file binario, chiamato "VOTI.DAT". Ciascun record di tale file contiene il nome ed il cognome dello studente, il codice del corso (un semplice intero) e la votazione ottenuta nell'esame.

Il nome ed il cognome dei candidati agli esami di laurea sono contenuti in un file di testo, chiamato "CANDIDATI.TXT", che contiene anche, all'inizio, il numero di candidati all'esame di laurea.

Si scriva un programma C per la segreteria dell'Università "La Sorbolona" che:

1. A partire dal file "STUDENTI.DAT", crei un vettore V di dimensioni opportune contenente le informazioni sui soli studenti laureandi (ovvero sugli studenti il cui nome e cognome sono contenuti nel file "CANDIDATI.TXT"). Per il confronto tra stringhe 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.
2. Utilizzando il vettore V, provveda a scrivere, su un file binario chiamato "LAUREANDI.DAT" le informazioni (nome, cognome, matricola e media voti) relative ai soli studenti laureandi. A tal scopo, si supponga l'esistenza di una funzione `media` che, dati il nome ed il cognome di uno studente, acceda al file "VOTI.DAT" e restituisca la media voti dello studente.
3. Letto un numero di matricola da tastiera, controlli se tale studente è candidato all'esame di laurea o meno e, in caso affermativo, ne stampi la media voti.
4. Si scriva il codice della funzione `media`. Le informazioni sui voti non devono essere memorizzate in strutture interne, ma vanno lette volta per volta dal file "VOTI.DAT". Per il confronto tra stringhe 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

Si illustri il concetto di variabile in C.

### Soluzione Esercizio 1

Rappresentazione 21:

$$(21)_{10} = (10101)_2$$

rappr. interna = **0001 0101**

Rappresentazione 5:

$$(5)_{10} = (101)_2$$

rappr. interna = **0000 0101**

Risultato **21 \* 5**:

$$\begin{array}{r} 10101 \quad \times \\ \underline{101} \quad = \\ 10101 \\ 1010100 \\ \hline 1101001 \end{array}$$

Risultato = 0110 1001 = (105)<sub>10</sub>

Risultato **43.625 - 105**:

Normalizzazione del risultato: 0.1101001 × 2<sup>7</sup>

Incolonnamento del numero più piccolo: 0.01010111 × 2<sup>7</sup>

Nessun errore di incolonnamento!

$$\begin{array}{r} 0.1101001 \times 2^7 - \\ \underline{0.01010111 \times 2^7} = \\ 0.01111011 \times 2^7 \end{array}$$

Normalizzando: - 0.11110110 × 2<sup>6</sup>

**Errore di cancellazione!**

byte esponente = **0000 0110**

byte mantissa = **1111 0110**

risultato: (- 111101.1)<sub>2</sub> = (- 61.5)<sub>10</sub>

risultato atteso: (- 61.375)<sub>10</sub>

### Soluzione Esercizio 2

Il risultato del programma è il seguente:

**0 1 2 4 3 2 1**

Infatti, alla prima invocazione della funzione F (i=7) viene calcolata la divisione intera tra gli ultimi due elementi del vettore V (2/1 = 2) ed il risultato viene posto nel 3° elemento del vettore (7/2-1 = 2); alla seconda invocazione della funzione F (i=5) viene calcolata la divisione intera tra il quarto ed il quinto elemento del vettore V (4/3 = 1) ed il risultato viene posto nel 2° elemento del vettore (5/2-1 = 1); alla terza invocazione della funzione F (i=3) viene calcolata la divisione intera tra il secondo ed il terzo elemento del vettore V (1/2 = 0) ed il risultato viene posto nel 1° elemento del vettore (3/2-1 = 0). I valori degli elementi del vettore sono quindi quelli indicati.

### Soluzione Esercizio 3

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
typedef struct { char nome[20], cognome[30]; int matricola; } studente;
typedef struct { char nome[20], cognome[30]; int codice, voto; } esame;
float media(char nome[], char cognome[]) {
    FILE *fp=fopen("VOTI.DAT", "rb");
    int N=0;
    float media=0;
    esame e;
    while(!feof(fp)) {
        fread(&e, sizeof(esame), 1, fp);
        if(!strcmp(e.cognome, cognome) && !strcmp(e.cognome, cognome)) {
            media+=e.voto;
            N++;
        }
    }
    return media/N;
}
main() {
    FILE *fp=fopen("CANDIDATI.TXT", "rt");
    int N, i=0, valore;
    studente *V;
    fscanf(fp, "%d", &N);
    V=(studente *)malloc(N*sizeof(studente));
    for(i=0; i<N; i++) {
        FILE *fps=fopen("STUDENTI.DAT", "rb");
        studente s;
        fscanf(fp, "%s %s", V[i].nome, V[i].cognome);
        while(!feof(fps)) {
            fread(&s, sizeof(studente), 1, fps);
            if(!strcmp(V[i].nome, s.nome) && !strcmp(V[i].cognome, s.cognome)) {
                V[i].matricola=s.matricola;
                break;
            }
        }
        fclose(fps);
    }
    fclose(fp); /* fine domanda 1 */
    fp=fopen("LAUREANDI.DAT", "wb");
    for(i=0; i<N; i++) {
        float m=media(V[i].nome, V[i].cognome);
        fwrite(&V[i], sizeof(studente), 1, fp);
        fwrite(&m, sizeof(float), 1, fp);
    }
    fclose(fp); /* fine domanda 2 */
    scanf("%d", &valore);
    fp=fopen("LAUREANDI.DAT", "rb");
    while(!feof(fp)) {
        studente s;
        float m;
        fread(&s, sizeof(studente), 1, fp);
        fread(&m, sizeof(float), 1, fp);
        if(s.matricola==valore) {
            printf("%d\n", m);
            break;
        }
    }
    fclose(fp); /* fine domanda 3 */
    free(V);
}
```