

**Fondamenti di Informatica - Ing. Civile/Edile - Dott. Penzo**  
**Soluzione compito B - 11/06/2001**

**Esercizio 1**

Per i passaggi intermedi vedere le dispense.

12 in base 2: 1100  
0.4 in base 2: .01100110....

12.4 in base 2: 1100.01100110....  
normalizzando: .11000110 mantissa e 00000100 esponente  
rappresentaz. : .01000110 mantissa e 00000100 esponente

24 in base 2: 11000  
7 in base 2: 00000111  
-7 in complemento a 2: 11111001

24 - 7 = 00010001 = 17 in base 10  
17 in floating point normalizzato: .10001000 mantissa e 00000101 esponente  
17 in rappresentazione interna: .00001000 mantissa e 00000101 esponente

12.4 incolonnato all'esponente maggiore:  
.01100011 mantissa e 00000101 esponente

somma 12.4 + 17: .01100011  
.10001000  
-----  
.11101011 mantissa e 00000101 esponente

ovvero: 11101.011 che in decimale vale  $16 + 8 + 4 + 1 + 0.25 + 0.125 = 29.375$   
(invece di 29.4)

**Esercizio 2**

La prima printf stampa il valore restituito dalla funzione F. Il valore stampato è 7 perché il ciclo for nella funzione F termina quando l'indice i supera o eguaglia il valore 5. Poiché i assume solo valori pari ( $i += 2$ ) il ciclo terminerà quando i assumerà valore 6 e verrà restituito al programma chiamante il valore di  $i+1$ , ossia 7, tramite l'istruzione return.

Poiché il vettore V viene passato per indirizzo a F, la funzione lo modifica e il ciclo for del main stampa i valori modificati per gli indici che vanno da 1 a 3 compresi, ossia: 2, 4, 4 (verticalmente).

### Esercizio 3

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

typedef struct {
    char nome[20];
    char cognome[20];
    char scuderia[20];
    int punteggio;
} pilota;

void bubblesort(pilota vett[], int inizio, int fine);
/* attenzione: l'ordinamento deve essere effettuato sulla scuderia */

void inTesta(char filename[20], char scud[20]);

int numelem = 0;

main()
{
    pilota elem;
    int i;
    pilota V[MAX_PILOTI];
    char scuderia[20];
    FILE *fp;

    fp = fopen("PILOTI.DAT","rb");
    if (fp == NULL) printf("Errore di apertura file PILOTI.DAT");
    else{
        while(!feof(fp)){
            fread(&elem, sizeof(pilota),1,fp);
            if (elem.punteggio > 0){
                strcpy(V[numelem].nome, elem.nome);
                strcpy(V[numelem].cognome, elem.cognome);
                strcpy(V[numelem].scuderia, elem.scuderia);
                V[numelem++].punteggio = elem.punteggio;
            }
        }
        fclose(fp);

        max = 0;
        for (i=1; i< numelem; i++)
            if (V[max].punteggio < V[i].punteggio)
                max = i;
        printf("Pilota in testa: %s %s con punti: %d\n", V[max].nome,
            V[max].cognome, V[max].punteggio);
    }
}
```

```

    bubblesort(V,0,numelem);
    fp = fopen("PILOTI.DAT","wb");
    for (i = 0; i < numelem; i++)
        fwrite(&V[i], sizeof(pilota),1,fp);
    fclose(fp);
    inTesta("PILOTI.DAT", scuderia);
    printf("Scuderia in testa alla classifica: %s", scuderia);

} /* else */
} /* main */

void inTesta(char filename[20], char scud[20])
{
    FILE *f;
    pilota primo, elem;
    int punteggio, maxpunteggio;

    f = fopen(filename, "rb");
    if (f == NULL) printf("Errore di apertura file");
    else {
        maxpunteggio = 0;
        fread(&primo, sizeof(pilota), 1, f);
        punteggio = primo.punteggio;
        while(!feof(f)){
            fread(&elem, sizeof(pilota),1,fp);
            if (!strcmp(elem.scuderia, primo.scuderia))
                punteggio += elem.punteggio;
            else {
                if(punteggio > maxpunteggio){
                    maxpunteggio = punteggio;
                    strcpy(scud, primo.scuderia);
                }
                primo = elem;
                punteggio = elem.punteggio;
            }
        }
    }
    fclose(f);
    return;
}

```