

**Fondamenti di Informatica - Ing. Civile/Edile - Dott. Penzo**  
**Soluzione compito B - 19/10/2001**

**Esercizio 1**

- 1) Con il metodo delle divisioni successive, al numero 135 in base 10 corrisponde il numero 10000111 in base 2. Per la conversione nelle basi 8 e 16 si possono applicare due metodi: nuovamente il metodo delle divisioni successive a partire da 135 oppure il raggruppamento in blocchi di 3 e di 4 cifre, rispettivamente per la base 8 e la base 16, del numero binario ottenuto in precedenza. I numeri che si ottengono sono: 207 in base 8 e 87 in base 16.
- 2) Occorre eseguire il procedimento inverso del complemento a 2 di un numero binario: sottrarre 1 al numero iniziale e complementare a 1 tutte le cifre. Si ottiene così il modulo del numero negativo iniziale. La conversione in base 10 di tale modulo consente di determinarne il valore decimale. Dal numero 11001010 sottraendo 1 si ottiene 11001001 che, complementato a 1, corrisponde a 00110110, ossia 54. Il numero iniziale equivale quindi a -54 in base 10.

**Esercizio 2**

Le istruzioni contenute all'interno del ciclo while del main vengono eseguite 3 volte. Infatti: per  $i = N - 1 = 4$  si ha  $B[4] = 3$  e  $4 > 3$  (la condizione del while è soddisfatta); per  $i = 3$  si ha  $B[3] = 2$  e  $3 > 2$ ; per  $i = 2$  si ha  $B[2] = 1$  e  $2 > 1$ ; infine per  $i = 1$  si ha  $B[1] = 2$  e  $1 > 2$  non è soddisfatta e si esce dal ciclo.

Per ogni iterazione viene chiamata la funzione F sul vettore V (che è l'unico che viene modificato!) con secondo parametro pari a 3, 2 e 1, rispettivamente nella prima, nella seconda e nella terza iterazione (per quanto detto sopra).

La prima esecuzione di F ha quindi  $j = 3$  e il for interno viene eseguito 4 volte: per  $i = 0$  fino a 3, dopodichè si esce perché per  $i = 4$  si ha  $4 \leq 3$  che risulta falso. Per  $i = 0$   $V[0]$  (il parametro formale è B ma quello attuale è V, quindi si agisce su V) assume il valore  $V[0] + (0 * 3) = 0$ .

Per  $i = 1$  a  $V[1]$  viene assegnato  $V[1] + (1 * 3) = 3$ . Per  $i = 2$  a  $V[2]$  viene assegnato  $V[2] + (2 * 3) = 6$ . Per  $i = 3$  a  $V[3]$  viene assegnato  $V[3] + (3 * 3) = 9$ . All'uscita del ciclo for  $i$  vale 4  
istruzione di return viene decrementato il valore di  $i$  ( $--i$ ) per cui viene restituito  $V[3]$  ossia 9. La prima printf del main stampa quindi il valore 9. Le rimanenti iterazioni del while nel main si comportano similmente, con le chiamate alla funzione F con  $j = 2$  e con  $j = 1$ , modificando ulteriormente il vettore V. Nel complesso il programma stampa:

9	10	6	(valori della funzione F)	
3	2	1	2	3 (vettore B inalterato)
0	6	10	9	0 (vettore V modificato)

### Esercizio 3

```
#include<stdio.h>

#define N 30

typedef struct {
    char nome[20];
    char presidente[30];
    int punteggio;
} squadra;

typedef struct {
    char *squadra_casa;
    char *squadra_ospite;
    int goal_casa;
    int goal_ospite;
} partita;

void GeneraFile(squadra C[], int n);
void AggiornaCampionato(char filename[], int num, squadra C[]);
void AggiornaVettore(squadra C[], int num, char *nome, int punti);
void Campione(squadra C[], int num, int *punti_campione);

main()
{
    int numelem = 0;
    squadra CAMPIONATO[N];
    int punti_campione;

    /* Fase di caricamento del vettore, non richiesta dal testo */
    /* si assuma che il numero di elementi contenuti sia numelem */

    GeneraFile(CAMPIONATO,numelem);
    AggiornaCampionato("PARTITE.DAT",numelem,CAMPIONATO);
    Campione(CAMPIONATO,numelem,&punti_campione);
}

void GeneraFile(squadra C[], int n)
{
    FILE *fp;
    int i = 0;

    fp = fopen("SQUADRE.TXT","w");
    if (fp == NULL) printf("Errore di apertura file!\n");
    else while(i < n){
        fprintf(fp,"%s%s%d",C[i].nome,C[i].presidente,C[i].punteggio);
        i++;
    }
    fclose(fp);
}

void AggiornaCampionato(char filename[], int num, squadra C[])
```

```

{
FILE *fp;
partita p;

fp = fopen(filename, "rb");
if (fp == NULL) printf("Errore di apertura file!\n");
else while(!feof(fp)){
    fread(&p,sizeof(partita),1,fp);
    if (p.goal_casa > p.goal_ospite)
        AggiornaVettore(C,num,p.squadra_casa,3);
    else if (p.goal_casa < p.goal_ospite)
        AggiornaVettore(C,num,p.squadra_ospite,3);
    else{
        AggiornaVettore(C,num,p.squadra_casa,1);
        AggiornaVettore(C,num,p.squadra_ospite,1);
    }
}
}

void AggiornaVettore(squadra C[], int num, char *nome, int punti)
{
int i=0;
int trovato = 0;

while((i < num) && !trovato){
    if (!strcmp(C[i].nome, nome)){
        trovato = 1;
        C[i].punteggio += punti;
    }
    i++;
}
}

void Campione(squadra C[], int num, int *punti_campione)
{
MinimumSort(C,num); /* procedura di sort per minimi successivi */
*punti_campione = C[0].punteggio;
}

```