

Fondamenti di Informatica - Ing. Civile/Edile - Dott. Penzo
Soluzione compito B - 07/12/2001

Esercizio 1

Per i passaggi intermedi vedere le dispense.

29 in base 2: 00011101
0.75 in base 2: .11000000....

25.25 in base 2: 11101.110000....
normalizzando: .11101110 mantissa e 00000101 esponente
rappresentaz. : .01101110 mantissa e 00000101 esponente

14 in base 2: 00001110
4 in base 2: 00000100
-4 in complemento a 2: 11111100

14 - 4 = 00001010 = 10 in base 10
10 in floating point normalizzato: .10100000 mantissa e 00000100 esponente
10 in rappresentazione interna: .00100000 mantissa e 00000100 esponente

10 incolonnato all'esponente maggiore:
.01010000 mantissa e 00000101 esponente

differenza 29.75 - 10: .11101110
 .01010000

 .10011110 mantissa e 00000101 esponente

ovvero: 10011.110 che in decimale vale $19 + 0.75 = 19.75$ (nessun errore)

Esercizio 2

Punto 1)

Ciclo for:

```
#include<stdio.h>
#define N 5

main{
int i;
for(i=1;i<N+1;i++){
printf("Iterazione numero %d\n", i+1);
printf("Iterazioni mancanti: %d\n", N-i);
}
}
```

Questo è l'elemento numero 1
Numero di elementi rimasti: 5
Questo è l'elemento numero 2
Numero di elementi rimasti: 4
Questo è l'elemento numero 3
Numero di elementi rimasti: 3
Questo è l'elemento numero 4
Numero di elementi rimasti: 2
Questo è l'elemento numero 5
Numero di elementi rimasti: 1

Punto 3)

Se N valesse 0, il programma non stamperebbe nulla, né con l'istruzione for, né con l'istruzione while, né con l'istruzione do-while.

Esercizio 3

```
#include<stdio.h>
#include<string.h>
#define MAX 1000
#define EURO 1936.27

typedef struct {
    int cod_prodotto;
    char descrizione[20];
    int num_reparto;
    int prezzo;
    int cod_rivenditore;
} prodotto;

typedef enum{false, true} bool;

void CaricaVettore(char *filename, char *descr, prodotto V[], int *numelem);
void GeneraScontati(char *filename, prodotto V[], int nelem, prodotto S[], int *n);
int RivenditoreMigliore(prodotto S[], int num, char *descr);

main()
{
    int prezzo, num, numscontati, codriv;
    prodotto R[MAX],SCONTATI[MAX];
    char descrizione[20];

    printf("Inserire la descrizione del prodotto: ");
    scanf("%s", descrizione);
    CaricaVettore("PRODOTTI.DAT", descrizione, R, &num);
    GeneraScontati("PROMOZIONI.TXT",R,num,SCONTATI,&numscontati);

    printf("Inserire il codice del rivenditore: ");
    scanf("%d", codriv);
    prezzo = PrezzoMigliore(SCONTATI, numscontati, codriv);
    printf("Il prezzo del prodotto più economico venduto dal rivenditore con codice %d è: %d",
    codriv, prezzo);
}

void CaricaVettore(char *filename, char *descr, prodotto V[], int *numelem)
{
    FILE *fp;
    prodotto p;
    int i=0;

    fp = fopen(filename, "rb");
    if (fp == NULL) printf("Errore di apertura file");
```

```

else {
    while(!feof(fp)){
        fread(&p,sizeof(prodotto),1,fp);
        if (!strcmp(p.descrizione,descr)){
            V[i].cod_prodotto = p.cod_prodotto;
            strcpy(V[i].descrizione,p.descrizione);
            V[i].num_reparto = p.num_reparto;
            V[i].prezzo = p.prezzo;
            V[i].cod_rivenditore = p.cod_rivenditore;
            i++;
        } /* end if */
    } /* end while */
} /* end else */
*numelem = i;
fclose(fp);
}

void GeneraScontati(char *filename, prodotto V[], int nelem, prodotto S[], int *n)
{
    FILE *fp;
    int codprodotto, sconto,i,j=0;
    bool trovato;

    fp = fopen(filename,"r");
    if (fp == NULL) printf("Errore di apertura file");
    else {
        while(!feof(fp)){
            fscanf(fp,"%d%d",&codprodotto,&sconto);
            if (sconto == 50){
                trovato = false;
                i = 0;
                while(i < nelem && !trovato){
                    if (V[i].cod_prodotto == codprodotto){
                        trovato = true;
                        S[j].cod_prodotto = V[i].cod_prodotto;
                        strcpy(S[j].descrizione, V[i].descrizione);
                        S[j].num_reparto = V[i].num_reparto;
                        S[j].prezzo = (V[i].prezzo - V[i].prezzo *
sconto/100)/EURO;

                        S[j].cod_rivenditore = V[i].cod_rivenditore;
                        j++;
                    } /* end if */
                    i++;
                } /* end while */
            } /* end if */
        } /* end while */
    }
}

```

```
    }/* end else */  
    fclose(fp);  
    *n = j;  
}
```

```
int PrezzoMigliore(prodotto S[], int num, int codriv)  
{  
    int minprezzo,i=0;  
    bool trovato = false;
```

```
    /* cerco il primo rivenditore del prodotto richiesto */  
    while(i < num && !trovato){  
        if (S[i].cod_rivenditore == codriv){  
            trovato = true;  
            minprezzo = S[i].prezzo;  
        } /* end if */  
        i++;  
    } /* end while */
```

```
    /* proseguo alla ricerca di eventuali altri prodotti forniti dallo stesso rivenditore e in offerta  
    migliore */  
    while(i < num){  
        if (S[i].cod_rivenditore == codriv && S[i].prezzo < minprezzo)  
            minprezzo = S[i].prezzo;  
        i++;  
    }  
    return(minprezzo);  
}
```