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

Esercizio 1
Per i passaggi intermedi vedere le dispense.

25 in base 2: 00011001
0.25 in base 2: .01000000....

25.25 in base 2: 11001.010000....
normalizzando: .11001010 mantissa e 00000101 esponente
rappresentaz. : .01001010 mantissa e 00000101 esponente

12 in base 2: 00001100
3 in base 2: 00000011
-3 in complemento a 2: 11111101

12 - 3 = 00001001 = 9 in base 10
9 in floating point normalizzato: .10010000 mantissa e 00000100 esponente
9 in rappresentazione interna: .00010000 mantissa e 00000100 esponente

9 incolonnato all'esponente maggiore:
.01001000 mantissa e 00000101 esponente

differenza 25.25 - 9: .11001010
.01001000

.10000010 mantissa e 00000101 esponente

ovvero: 10000.010 che in decimale vale 16 + 0.25 = 16.25 (nessun errore)

Esercizio 2

Punto 1)
Ciclo while:

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

main{
int i;
i = 0;
while(i < N){
     printf("Iterazione numero %d\n", i+1);
     printf("Iterazioni mancanti: %d\n", N-i);
     i++;}
}
```

Ciclo do-while:

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

main{
int i;
i = 0;
if (i < N)
     do{
          printf("Iterazione numero %d\n", i+1);
          printf("Iterazioni mancanti: %d\n", N-i);
          i++;
     }while(i<N);
}
```

Punto 2)

L'istruzione while equivalente al for è ottenuta tramite l'opportuna inizializzazione (a zero dato che il for parte da zero) della variabile di ciclo i e tramite il controllo sulla condizione di ciclo, secondo argomento dell'istruzione for, espressa come condizione dell'istruzione while. L'incremento della variabile di ciclo deve avvenire all'interno del corpo del while.
L'istruzione do-while, per essere considerata equivalente all'istruzione for, deve essere preceduta da un controllo sulla condizione di ingresso al ciclo (istruzione if), poiché in alcuni casi (ad esempio se N valesse 0) il do-while eseguirebbe il corpo del ciclo mentre il for e il while non eseguirebbero nessuna istruzione. Non sarebbero quindi equivalenti.

Il programma termina quando i vale 5 (5 non è minore di N=5) e l'output del programma è:

Iterazione numero 1
Iterazioni mancanti: 5
Iterazione numero 2
Iterazioni mancanti: 4
Iterazione numero 3
Iterazioni mancanti: 3
Iterazione numero 4
Iterazioni mancanti: 2
Iterazione numero 5
Iterazioni mancanti: 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, int numrep, 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 numreparto, num, numscontati, codriv;
prodotto R[MAX],SCONTATI[MAX];
char descrizione[20];

printf("Inserire il numero di reparto: ");
scanf("%d", &numreparto);
CaricaVettore("PRODOTTI.DAT", numreparto, R, &num);
GeneraScontati("PROMOZIONI.TXT",R,num,SCONTATI,&numscontati);
printf("Inserire la descrizione: ");
scanf("%s", descrizione);
codriv = RivenditoreMigliore(SCONTATI, numscontati, descrizione);
printf("Il codice del rivenditore che fornisce il prodotto %s più economico è: %d",
descrizione, codriv);
}

void CaricaVettore(char *filename, int numrep, 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 (p.num_reparto == numrep){
                        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 > 40){
                        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 RivenditoreMigliore(prodotto S[], int num, char *descr)
{
int minprezzo,codriv,i=0;
bool trovato = false;

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

/* proseguo alla ricerca di eventuali altri rivenditori del prodotto con offerte migliori */
while(i < num){
        if (!strcmp(S[i].descrizione, descr) && S[i].prezzo < minprezzo){
                minprezzo = S[i].prezzo;
                codriv = S[i].cod_rivenditore;
        }
        i++;
}
return(codriv);
}
```