

Fondamenti di Informatica - Ing. Civile/Edile - Dott. Penzo
Soluzione compito A - 20/09/2001

Esercizio 1

Per i passaggi intermedi vedere le dispense.

5 in base 2: 00000101
0.27 in base 2: .010001....

5.27 in base 2: 101.010001....
normalizzando: .10101000 mantissa e 00000011 esponente
rappresentaz. : .00101000 mantissa e 00000011 esponente

48 in base 2: 00110000
37 in base 2: 00100101
-37 in complemento a 2: 11011011

48 - 37 = 00001011 = 11 in base 10
11 in floating point normalizzato: .10110000 mantissa e 00000100 esponente
11 in rappresentazione interna: .00110000 mantissa e 00000100 esponente

5.27 incolonnato all'esponente maggiore:
 .01010100 mantissa e 00000100 esponente

5.27 - 11 = - (11 - 5.27)
differenza 11 - 5.27: .10110000
 .01010100

 .01011100 mantissa e 00000100 esponente

normalizzando: .10111000 mantissa e 00000011 esponente
ovvero: 101.11000 che in decimale vale $4 + 1 + 0.5 + 0.25 = 5.75$ quindi cambiando il segno: -5.75 (invece di -5.73)

Esercizio 2

La prima printf stampa il valore restituito dalla funzione F. Il valore stampato è 8 perché il ciclo for nella funzione F termina quando l'indice i eguaglia il valore n che corrisponde al parametro inizializzato a 4. Quindi il ciclo terminerà quando i assumerà valore 4 e la somma di tale valore con il valore di n (che vale sempre 4) verrà restituita al programma chiamante tramite l'istruzione return.

Poiché il vettore A viene passato per indirizzo a F, la funzione lo modifica nel modo seguente: A[0] = 3, A[1] = 3, A[2] = 0, A[3] = 2, A[4] non viene modificato. I nuovi valori contenuti in A corrispondono ai resti delle divisioni intere dei valori originali con il valore 4 (valore del parametro n). Il ciclo for del main stampa i seguenti valori (senza modificare il vettore!): 2, -2, -1, -3, corrispondenti rispettivamente a A[i] - i per i valori di i da 1 a 4 (escludendo quindi il primo elemento).

Esercizio 3

```
#include <stdio.h>
#define N 100
typedef enum {false,true} bool;
typedef struct elem{
    int codice;
    char nome[20];
    char vitigno[20];
    char tipo[6];
    bool frizzante;
    int gradazione;
} vino;

int PiuCorposi(vino V[], int i);
int PrimoDisponibile(char nomefile[], char nome[20], int nb);
void GestioneOrdini(vino V[], int i, char nome[20], int nb, int *imp);

main()
{
    FILE *f1;
    vino v, ROSSI[N];
    char nome[20];
    int i=0, numvini, importo=0, numbott;

    f1 = fopen("VINI.DAT","rb");
    if (f1 == NULL) printf("Errore di apertura file\n");
    else {
        while(!feof(f1))
        {
            fread(&v, sizeof(vino), 1, f1);
            if(!strcmp(v.tipo, "rosso") && ((!v.frizzante) || (v.gradazione > 9)))
            {
                ROSSI[i].codice = v.codice;
                strcpy(ROSSI[i].nome, v.nome);
                strcpy(ROSSI[i].vitigno, v.vitigno);
                strcpy(ROSSI[i].tipo, v.tipo);
                ROSSI[i].frizzante = v.frizzante;
                ROSSI[i].gradazione = v.gradazione;
                i++;
            } /* end if */
        } /* end while */
        numvini = PiuCorposi(ROSSI,i);
    }
}
```

```

        printf("Inserire il nome del vino da ordinare:");
        scanf("%s",nome);
        printf("Inserire il numero di bottiglie richieste:");
        scanf("%d", &numbott);
        GestioneOrdini(ROSSI, i, nome, numbott, &importo);
        printf("L'importo totale dell'ordine e` pari a: %d", importo);
    } /* end else */
fclose(f1);
} /* end main */

int PiuCorposi(vino V[], int n)
{
    int i, max, numvini =0;

    max = V[0].gradazione;
    for(i=1; i < n; i++)
    {
        if (V[i].gradazione > max)
            max = V[i].gradazione;
    } /* end for */
    printf("I vini con gradazione alcolica piu` alta sono: \n");
    for (i=0; i < n; i++)
    {
        if (V[i].gradazione == max)
        {
            printf("Il vino %s con codice %d\n", V[i].nome, V[i].codice);
            numvini++;
        } /* end if */
    } /* end for */
    return(numvini);
}

int PrimoDisponibile(char nomefile[], char nome[20], int numbottiglie)
{
    FILE *fp1, *fp2;
    int cod, numbott, prezzo, price = 0;
    bool trovatocodice=0, trovatovino=0;
    vino v;

    fp1 = fopen(nomefile,"r");
    fp2 = fopen("VINI.DAT","rb");
    if(fp1 == NULL || fp2 == NULL) printf("Errore di apertura file\n");
    else{
        while(!feof(fp1) && !trovatovino)
        {
            fscanf(fp1,"%d%d%d",&cod,&numbott,&prezzo);

```

```

if(numbott >= numbottiglie)
{
    rewind(fp2);
    while(!feof(fp2) && !trovatocodice)
    {
        fread(&v, sizeof(vino), 1, fp2);
        if(v.codice == cod) trovatocodice = 1;
        if(!strcmp(v.nome,nome))
        {
            trovatovino = 1;
            price = prezzo;
        } /* end if */
    } /* end while */
} /* end if */
return(price);
}

void GestioneOrdini(vino V[], int i, char nome[20], int nb, int *imp)
{
int prezzo;

prezzo = PrimoDisponibile("VENDEMMIA.TXT", nome, nb);
if (prezzo > 0)
    *imp = prezzo * nb;
else printf("Non e` possibile effettuare l'ordine\n");
}

```