

5 Switch, typedef e matrici

Soluzioni

Soluzione dell'esercizio 5.1

```
#define LEN 256
#define MAX_ESAMI 20
#define LEN_CODICE 6

#include <stdio.h>
#include <string.h>

//tipo stringa generico
typedef char string_t[LEN+1];

//tipo data: ad esempio, 13/06/2012
typedef struct data {
    int giorno;
    int mese;
    int anno;
} data_t;

//tipo esame
typedef struct esame {
    int voto;
    data_t data;
    string_t nome;
    int codice;
} esame_t;

//definisco il libretto
typedef esame_t libretto_t[MAX_ESAMI];

int main()
{
    /* DICHIARAZIONI */
    //operazione
    char op;
    int inserimento, stampa, ricerca;

    //flag di validita' operazione
    int op_valida;

    //numero esami
    int n_esami = 0;

    //indice
    int i;

    //variabili accessorie
    int codice;
    int trovato = 0;

    //definisco una variabile libretto di tipo "libretto_t"
    libretto_t libretto;

    //ciclo do-while esterno per selezione operazione
    do {
        //variante fflush(stdin), che a volte fallisce
        while(getchar() != '\n');
```

```

do {
    printf("\n"
           "[i] inserimento nuovo esame\n"
           "[r] ricerca per codice\n"
           "[p] stampa libretto\n"
           "[x] uscita\n"
           "\n");
    scanf("%c", &op);

    fflush(stdin);

    op_valida = (op == 'i' || op == 'r' || op == 'p' || op == 'x');

    if (!op_valida)
        printf("Operazione non valida [i/r/p/x]: ");
} while(!op_valida);

inserimento = 0;
ricerca = 0;
stampa = 0;

//selezione operazione
switch (op) {
    case 'i':
        if (n_esami < MAX_ESAMI)
            inserimento = 1;
        else
            printf("Numero massimo di esami raggiunto\n");
        break;
    case 'r':
        ricerca = 1;
        break;
    case 'p':
        stampa = 1;
        break;
    //default, uscita :)
}

//lo switch e l'inizializzazione delle flag a zero mi assicurano
//che a questo punto, solo una delle operazioni sia selezionata

if (inserimento) {
    printf("Inserimento dell'esame numero %d\n", n_esami+1);

    //acquisizione del codice dell'i-esimo esame
    do {
        printf("Codice: ");
        scanf("%d", &libretto[n_esami].codice);
        fflush(stdin);
    } while (libretto[n_esami].codice < 0 || libretto[n_esami].codice > 999999)
        ;

    //acquisizione data del corso
    printf("Data in formato GG/MM/AAAA: ");
    scanf("%d/%d/%d",
          &libretto[n_esami].data.giorno,
          &libretto[n_esami].data.mese,
          &libretto[n_esami].data.anno);
    fflush(stdin);

    //acquisizione nome esame
    printf("Nome corso: ");

```

```

scanf("%s", libretto[n_esami].nome);

fflush(stdin);

do {
    printf("Voto: ");
    scanf("%d", &libretto[n_esami].voto);
    fflush(stdin);
} while(libretto[n_esami].voto < 18 || libretto[n_esami].voto > 30);

n_esami++;
}

if (ricerca) {
    do {
        printf("Codice da cercare: ");
        scanf("%d", &codice);
        fflush(stdin);
    } while (codice < 0 || codice > 999999);

    trovato = 0;

    for (i = 0; i < n_esami && !trovato; i++)
        if (libretto[i].codice == codice) {
            trovato = 1;
            printf("\n"
                "Data: %d/%d/%d\n"
                "Codice: %d\n"
                "Nome corso: %s\n"
                "Voto: %d\n",
                libretto[i].data.giorno,
                libretto[i].data.mese,
                libretto[i].data.anno,
                libretto[i].codice,
                libretto[i].nome,
                libretto[i].voto);
        } else {
            printf("Esame non trovato\n");
        }
    }

    if (stampa) {
        for (i = 0; i < n_esami; i++)
            printf("Data: %d/%d/%d\n"
                "Codice: %d\n"
                "Nome corso: %s\n"
                "Voto: %d\n",
                libretto[i].data.giorno,
                libretto[i].data.mese,
                libretto[i].data.anno,
                libretto[i].codice,
                libretto[i].nome,
                libretto[i].voto);
    }
} while(op != 'x');
}

```

Soluzione dell'esercizio 5.2

```
#define D 10
```

```
#include <stdio.h>

typedef int matrix_t[D][D];

int main(void)
{
    matrix_t m;

    int i, j, d;

    //acquisizione dimensione effettiva
    do {
        printf("inserire il numero di righe e colonne\n");
        scanf("%d", &d);
    } while (d < 0 || d > d);

    //lettura valori
    for (i = 0; i < d; i++) {
        for (j = 0; j < d; j++) {
            printf("m[%d][%d] = ", i+1, j+1);
            scanf("%d", &m[i][j]);
        }
    }

    //scansione lungo la diagonale mentre non ci sono zeri
    for (i = 0; m[i][i] != 0 && i < d; i++);

    //se i < d significa che sono uscito prima dal ciclo
    if (i < d)
        printf("non ");

    printf("diagonale\n");

    return 0;
}
```