

ESERCIZIO

(è una variante dell'esercizio delle dispense "Esercizi-record, stringhe e funzioni" pag 5)

Scrivere un programma C che, date le seguenti dichiarazioni

```
typedef struct { int Matr;  
                char Facol;  
            } Studente;
```

```
Studente Studenti[10];
```

esegua le seguenti operazioni:

1) Riportare in output gli studenti la cui facoltà (campo Facol) è 'A' 'L' oppure 'I':

Studenti	Matr	Facol	Output
0	32	'I'	32 I
1	43	'M'	65 I
2	65	'I'	21 L
3	21	'L'	89 I
4	89	'I'	434 A
5	53	'M'	
6	123	'F'	
7	12	'M'	
8	434	'A'	
9	3	'M'	

2) Riportare in output il numero di studenti della facoltà di Ingegneria (campo Facol pari a 'I') con numero di matricola dispari. Nell'esempio precedente tale numero è 2.

3) Riportare le Facoltà con un solo studente, cioè le facoltà che compaiono una sola volta nel vettore Studenti, ovvero il cui valore è distinto dagli altri valori presenti nel vettore. Nell'esempio 'L' e 'F'.

4) Ordinare ed elencare (stampare a video) gli studenti in base alla facoltà e, a parità di facoltà, in base alla matricola.

5) Utilizzando l'ordinamento ottenuto al punto precedente, calcolare e stampare a video, per ogni facoltà, il numero di studenti. Nell'esempio si deve ottenere:

```
La Facolta A ha 1 studenti  
La Facolta F ha 1 studenti  
La Facolta I ha 3 studenti  
La Facolta L ha 1 studenti  
La Facolta M ha 1 studenti
```

Suggerimento: Per risolvere questo punto utilizzare la stessa tecnica (e quindi un algoritmo simile) a quello visto nell'esercizio 2.9 –pag 25 delle dispense con gli esercizi.

6) Considerando anche un vettore di Laureandi (i laureandi hanno la stessa struttura degli studenti):
Studente Studenti[6];

calcolare e riportare in output gli Studenti che non sono Laureandi, ovvero i record del vettore Studenti per i quali non esiste un record con la stessa matricola nel vettore Laureandi. Esempio

Laureandi	Matr	Facol	
0	43	'M'	
1	65	'I'	
2	21	'L'	123 F
3	89	'I'	32 I
4	3	'M'	12 M
5	51	'I'	53 M

Suggerimento: Per risolvere questo punto utilizzare la stessa tecnica (e quindi un algoritmo simile) a quello dell'esercizio Esercizio 2.1 – pag 23 delle dispense con gli esercizi.

L'ordine con il quale vengono presentati i record in output è irrilevante.

7) Considerando il vettore di Laureandi dichiarato in precedenza, calcolare e riportare in output gli Studenti di Ingegneria che sono anche Laureandi, ovvero i record del vettore Studenti con il campo Facol pari a 'I' per i quali esiste un record con la stessa matricola nel vettore Laureandi. Nell'esempio

Output

65 'I'
89 'I'

NOTE:

Non è necessario inizializzare e/o acquisire il vettore Studenti ed il vettore Laureandi

ESERCIZIO

```
#include <stdio.h>
#include <stdlib.h>

main(){
    typedef struct {
        int Matr;
        char Facol;
    } Studente;

    Studente Studenti[10] = /* inizializzazione */
    {{32, 'I'},
     {43, 'M'},
     {65, 'I'},
     {21, 'L'},
     {89, 'I'},
     {53, 'M'},
     {123, 'F'},
     {12, 'M'},
     {434, 'A'},
     {3, 'M'}}};

    Studente Laureandi[6] = /* inizializzazione */
    { {43, 'M'},
      {65, 'I'},
      {21, 'L'},
      {89, 'I'},
      {3, 'M'},
      {51, 'I'}}};

    int I,J;
    int NSD=0; /* studenti di ingegneria con matricola dispari*/
    int count;
    char Facolta;
    Studente temp;
    int trovato;

    /* domanda 1 */
    for(I=0;I<10;I++)
    if ( Studenti[I].Facol == 'A' || Studenti[I].Facol == 'L' ||
        Studenti[I].Facol == 'I')
        printf("Matricola : %d - Facolta' : %c\n",
                Studenti[I].Matr,Studenti[I].Facol);

    /* domanda 2 */
    for(I=0;I<10;I++)
        if (Studenti[I].Matr%2==1 && Studenti[I].Facol=='I')
            NSD++;
    printf("studenti di ingegneria con matricola dispari : %d \n",NSD);
```

```

/* domanda 3 */
printf("Facolta con un solo studente - soluzione non corretta \n");
for(I=0;I<10;I++){
    Facolta=Studenti[I].Facol;
    count=0;
    for(J=I+1;J<10;J++)
        if (Studenti[J].Facol==Facolta)
            count++;
    if (count==0)
        printf("%c \n",Facolta);
}

/* domanda 3 */
printf("Facolta con un solo studente - soluzione corretta \n");
for(I=0;I<10;I++){
    Facolta=Studenti[I].Facol;
    count=0;
    for(J=0;J<10;J++)
        if (Studenti[J].Facol==Facolta)
            count++;
    if (count==1)
        printf("%c \n",Facolta);
}

/* domanda 4 */
printf("\n Ordinamento con Bubble-Sort \n");
for(I=0;I<10-1;I++)
    for(J=0;J<10-I-1;J++)
        if (Studenti[J+1].Facol<Studenti[J].Facol ||
            Studenti[J+1].Facol == Studenti[J].Facol &&
            Studenti[J+1].Matr < Studenti[J].Matr ){
            temp= Studenti[J+1];
            Studenti[J+1]= Studenti[J];
            Studenti[J] = temp;
        }

printf("\n Studenti Ordinati \n");
for(I=0;I<10;I++)
    printf("Matricola : %d - Facolta' : %c\n",
           Studenti[I].Matr,Studenti[I].Facol);

/* domanda 5 */
count=1;
for (I=0;I< 10 -1;I++){
    if (Studenti[I+1].Facol == Studenti[I].Facol)
        count++;
    else {
        printf(" La Facolta %c ha %d studenti \n",Studenti[I].Facol,count);
        count=1;
    }
}
printf(" La Facolta %c ha %d studenti \n",Studenti[10-
1].Facol,count);

```

```

/* domanda 6 */
printf("\n Studenti che non sono laureandi \n");
for(I=0;I<10;I++){
    /* inizio la ricerca dello studente I-esimo in Laureandi */
    trovato=0;
    for(J=0;J<5;J++)
        if ( Studenti[I].Matr == Laureandi[J].Matr)
            trovato =1;
/* qui la ricerca in Laureandi e' terminata */

/* se studente I-esimo non è stato trovato in Laureandi lo stampo */
if (trovato==0)
    printf("Matricola : %d - Facolta' : %c\n",
           Studenti[I].Matr,Studenti[I].Facol);
}

/* domanda 7 */
printf("\n Studenti di Ingegneria che sono laureandi \n");
for(I=0;I<10;I++){
/* prima di tutto controllo che lo studente I-esimo sia di Ingegneria
*/
    if ( Studenti[I].Facol=='I') {

        /* inizio la ricerca dello studente I-esimo in Laureandi */
        trovato=0;
        for(J=0;J<5;J++)
            if ( Studenti[I].Matr == Laureandi[J].Matr)
                trovato =1;
/* qui la ricerca in Laureandi e' terminata */

/* se studente I-esimo è stato trovato in Laureandi lo stampo */
if (trovato==1)
    printf("Matricola : %d - Facolta' : %c\n",
           Studenti[I].Matr,Studenti[I].Facol);
    }
}
}

```