Il costruttore struct

Generalità

- Il costruttore di tipi **struct** permette di definire dei tipi strutturati aggregando informazioni eterogenee tra loro correlate.

- Una struttura viene definita mediante il costruttore struct nel seguente modo:

```
struct nome_struttura
{
        tipo_1          nome_campo_1;
        tipo_2          nome_campo_2;
        . . . . . .
        tipo_N          nome_campo_N;
};
```

dove *tipo_i* e *nome_campo_i* indicano, rispettivamente, il tipo e il nome dell'i-esimo campo della struttura.

Esempio: struct studente

- Volendo definire una struttura che mette in relazione le informazioni relative ad uno studente (*nome, cognome, matricola, numero esami superati, voto medio)* potremo usare il costruttore struct nel seguente modo:

```
struct studente
{
    char        cognome[20];
    char        nome[20];
    long        matricola;
    int         numero_esami;
    float       media;
};
```

Dichiarazione di una variabile di tipo struct

- La dichiarazione di una variabile di questo tipo è analoga a quella delle variabili di tipo predefinito C.

- La sintassi da utilizzare per la dichiarazione è la seguente:

```
struct nome_struttura elenco_variabili;
```

- Volendo ad esempio dichiarare una variabile di nome **stud1** di tipo struct studente scriveremo:

```
struct studente stud1;
```

- La dichiarazione della variabile deve essere preceduta dalla definizione della struttura.

Dichiarazione di una variabile di tipo struct

- In alternativa è possibile definire la struttura e dichiarare una o più variabile nel seguente modo:

```
struct nome_struttura
{
    tipo_1      nome_campo_1;
    tipo_2      nome_campo_2;
    . . . . . .
    tipo_N      nome_campo_N;
} elenco_variabili;
```

Dichiarazione di una variabile di tipo struct

- Per definire la struct studente e dichiarare tre variabili (stud1, stud2 e stud3), potremo scrivere:

```
struct studente
{
 char            nome[20];
 char            cognome[20];
 char            matricola[11];
 int             numero_esami;
 float           media;
} stud1,stud2;

struct studente stud3;
```

Accesso ai campi di una struttura

- Per accedere ai campi di una struttura si utilizza l'operatore punto nel seguente modo:

```
nome_variabile.nome_campo
```

- Volendo, ad esempio, accedere ai campi di una variabile di nome stud e di tipo struct studente scriveremo:

```
stud.cognome
stud.nome
stud.matricola
stud.numero_esami
stud.media
```

Assegnamento di struct

- Date due variabili dello stesso tipo possiamo copiare i campi di una variabile in quelli dell'altra variabile mediante un unico assegnamento.

Es. Per copiare i campi della variabile stud1 in quelli della variabile stud2 scriveremo:

```
stud2=stud1;
```

Vettore di strutture

- La dichiarazione di un vettore di strutture è analoga a quella dei vettori di tipi semplici. Essa viene fatta nel seguente modo:

```
struct nome_struttura  nome_vettore[dim];
```

- Volendo dichiarare un vettore di nome classe di 10 elementi di tipo struct studente scriveremo:

```
struct studente
{
char                nome[20];
char                cognome[20];
long                matricola;
int                 numero_esami;
float               media;
};
struct studente S[10];
```

Vettore di strutture

Per accedere ai singoli campi dell'i-esimo studente scriveremo:

```
S[i].nome;
S[i].cognome;
S[i].matricola;
S[i].numero_esami;
S[i].media;
```

Lettura e visualizzazione degli elementi di tipo struct studente di un vettore (1)


```
#include <stdio.h>
#define N 20
struct studente {
 char   nome[20];
 char   cognome[20];
 long   matricola;
 int    nmaterie;
 float  media;
};

main()
{int presenti,ind;
 struct studente S[N];

 printf("Indicare il numero di studenti \n");
 scanf("%d",&presenti);
```

Lettura e visualizzazione degli elementi di tipo struct studente di un vettore (2)


```
for(ind=0;ind<presenti;ind++)
 {printf("Cognome: ");
  scanf("%s",S[ind].cognome);
  printf("Nome: ");
  scanf("%s",S[ind].nome);
  printf("Matricola: ");
  scanf("%ld",&S[ind].matricola);
  printf("Numero di materie: ");
  scanf("%d",&S[ind].nmaterie);
  printf("Media: ");
  scanf("%f",&S[ind].media);
 }

for(ind=0;ind<presenti;ind++)
 {printf("Cognome: %s\n",S[ind].cognome);
  printf("Nome: %s\n",S[ind].nome);
  printf("Matricola: %s\n",S[ind].matricola);
  printf("Numero di materie: %d\n",S[ind].nmaterie);
  printf("Media: %f\n",S[ind].media);
 }
}
```

Lettura degli elementi di tipo struct studente di un vettore e successive ricerca sequenziale (1)


```
#include <stdio.h>
#define N 20
struct studente {
 char   cognome[20],nome[20];
 long   matricola;
 int    nmaterie;
 float  media;
};

main()
{int ind,trovato;
 long matricola;
 struct studente S[N];

for(ind=0;ind<N;ind++)
 {printf("Cognome: ");          scanf("%s",S[ind].cognome);
  printf("Nome: ");             scanf("%s",S[ind].nome);
  printf("Matricola: ");        scanf("%ld",&S[ind].matricola);
  printf("Numero materie: "); scanf("%d",&S[ind].nmaterie);
  printf("Media: ");            scanf("%f",&S[ind].media);
 }
```

Lettura degli elementi di tipo struct studente di un vettore e successive ricerca sequenziale (2)


```
 printf("Inserisci il numero di matricola: ");
 scanf("%ld",&matricola);
/* Ricerca */
 for(ind=0;ind<N)
  if(S[ind].matricola==matricola)
       break;

 if(ind<N)
 {printf("Cognome: %s\n",S[ind].cognome);
  printf("Nome: %s\n",S[ind].nome);
  printf("Matricola: %ld\n",S[ind].matricola);
  printf("Numero di materie: %d\n",S[ind].nmaterie);
  printf("Media: %f\n",S[ind].media);
 }
 else printf("Non e' presente");
}
```

Lettura degli elementi di tipo struct studente di un vettore e ordinamento mediante bubble sort (1)

Prof. G. Ascia

```
#include <stdio.h>
#define N 20
struct studente {
 char   cognome[20],nome[20];
 long   matricola;
 int    nmaterie;
 float  media;
};

main()
{int ind,sup;
 long matricola;
 struct studente S[N],aux;

for(ind=0;ind<N;ind++)
 {printf("Cognome: ");          scanf("%s",S[ind].cognome);
  printf("Nome: ");             scanf("%s",S[ind].nome);
  printf("Matricola: ");        scanf("%ld",&S[ind].matricola);
  printf("Numero materie: "); scanf("%d",&S[ind].nmaterie);
  printf("Media: ");            scanf("%f",&S[ind].media);
 }
```

Lettura degli elementi di tipo struct studente di un vettore e ordinamento mediante bubble sort (2)


```
 for(sup=N-1; sup>0;sup--)
    for(ind=0;ind<sup;ind++)
        if(S[ind+1].matricola<S[ind].matricola)
         { aux=S[ind+1];
           S[ind+1]=S[ind];
           S[ind]=aux;
         }

 for(ind=0;ind<N;ind++)
  {printf("Cognome: %s\n",S[ind].cognome);
   printf("Nome: %s\n",S[ind].nome);
   printf("Matricola: %ld\n",S[ind].matricola);
   printf("Numero di materie: %d\n",S[ind].nmaterie);
   printf("Media: %f\n",S[ind].media);
   }

}
```