

LE STRUTTURE



Le strutture o record

I record o strutture sono strutture dati statiche non omogenee cui si fa riferimento usando un unico identificatore.

Il **tipo** di dato struttura deve essere dichiarato, in modo da definire quanti e quali dati dovranno essere aggregati nella struttura.

Dichiarazione del tipo struttura

```
struct nome_struttura{
    tipo_variabile1 nome_variabile1;
    tipo_variabile2 nome_variabile2;
    ... };
```

Ogni singola variabile appartenente alla struttura è detta campo della struttura.

Dichiarazione del tipo struttura

```
struct studente{
    int matricola;
    char nome[20];
    char cognome[15];
    float media;
    char corso;};
```

| matricola | nome | cognome | media | corso |
|-----------|---------|---------|-------|-------|
| Intero | Stringa | Stringa | Float | Char |

Visibilità del tipo struttura

```
#include <stdio.h>
struct ... ← Ambiente globale.
                Il tipo struct è visibile ovunque nel
                programma.

main()
{ struct ...
  ... }

fun()
{ struct ...
  ... }
```

↳ Ambiente locale della funzione.
Il tipo struct è visibile nella funzione.

Dichiarazione di variabile di tipo struttura

La dichiarazione delle variabili di tipo struttura può essere fatta contemporaneamente alla dichiarazione del tipo

```
struct nome_struttura{
    tipo_variabile1 nome_variabile1;
    tipo_variabile2 nome_variabile2;
    ...; } s, t;
```

oppure separatamente

```
struct nome_struttura s, t;
```

Assegnazione valori a struttura

Per assegnare i valori ad una struttura è necessario assegnarli ai singoli campi che la compongono.

E' possibile assegnare i valori in modo diretto:

```
struct nome_struttura{
    tipo_variabile1 nome_variabile1;
    tipo_variabile2 nome_variabile2;
    ...; } s, t;
s.nome_variabile1=...;
s.nome_variabile2=...; .... Attenzione ai campi stringa!
```

Assegnazione valori a struttura

```
struct studente{
    int matricola;
    char nome[20];
    char cognome[15];
    float media;
    char corso;};
s.matricola=1234;
strcpy(s.nome, "Mario");
strcpy(s.cognome, "Rossi");
s.media=28.5;
s.corso='B';
```

L'ordine di assegnazione dei valori ai campi è indifferente.

Assegnazione valori a struttura

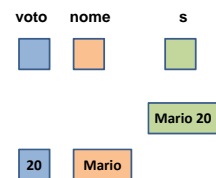
E' possibile assegnare i valori dei campi di una struttura acquisendoli da tastiera:

```
struct nome_struttura{
    tipo_variabile1 nome_variabile1;
    tipo_variabile2 nome_variabile2;
    ...; } s, t;
scanf("formato di controllo", &s.nome_variabile1);
scanf("formato di controllo", &s.nome_variabile2);
...
```

Assegnazione valori a struttura

Per acquisire da tastiera i valori dei campi di una struttura è possibile usare la funzione **sscanf**, che riceve una stringa e pone i vari campi (separati da uno spazio bianco) nelle variabili indicate.

```
main()
{int voto;
char nome[20], s[100];
gets(s);
sscanf(s, "%s %d", nome, &voto)}
```



Assegnazione valori a struttura

```
struct alunno{char nome[20];
int voto;};
main()
{struct alunno a;
char s[100];
gets(s);
sscanf(s, "%s %d",&a.nome, &a.voto)}
```

Visualizzazione valori di una struttura

Per visualizzare i valori di una struttura è necessario visualizzare i valori dei singoli campi.

```
struct nome_struttura{
    tipo_variabile1 nome_variabile1;
    tipo_variabile2 nome_variabile2;
    ...; }s;
printf("formato di controllo", s.nome_variabile1);
printf("formato di controllo", s.nome_variabile2);
...
```

Copia dei valori di una struttura

E' possibile effettuare la copia diretta dei valori contenuti in una struttura, dentro una seconda struttura, del medesimo tipo.

```
struct nome_struttura{
    tipo_variabile1 nome_variabile1;
    tipo_variabile2 nome_variabile2;
    ...; }s, t;

s=t;
```

Vettori di strutture

E' possibile generare array di tipo struttura.

Dopo aver dichiarato il tipo struttura si dichiara l'array:

```
struct nome_struttura nome_vettore[dimensione];
```

```
struct studente v[3];
```

| | matricola | nome | cognome | media | corso |
|---|-----------|---------|---------|-------|-------|
| 0 | Intero | Stringa | Stringa | Float | Char |
| 1 | | | | | |
| 2 | | | | | |

Assegnazione valori a vettori di strutture

E' necessario assegnare il valore ai singoli campi delle singole componenti.

```
for(i=0;i<N;i++)
{ scanf("%d",&v[i].matricola);
  scanf("%s",&v[i].nome);
  scanf("%s",&v[i].cognome);
  scanf("%f",&v[i].media);
  scanf("%c",&v[i].corso);}
```

Visualizzazione valori di vettori di strutture

E' necessario visualizzare il valore dei singoli campi delle singole componenti.

```
for(i=0;i<N;i++)
{ printf("%d",v[i].matricola);
  printf("%s",v[i].nome);
  printf("%s",v[i].cognome);
  printf("%f",v[i].media);
  printf("%c",v[i].corso);}
```

Puntatori a struttura

Dopo aver dichiarato il tipo struttura si dichiara il puntatore:

```
struct nome_struttura *nome_puntatore;
```

```
struct studente *p;
```

| matricola | nome | cognome | media | corso |
|-----------|---------|---------|-------|-------|
| Intero | Stringa | Stringa | Float | Char |

↑
p

Puntatori a struttura

E' possibile raggiungere i singoli campi della struttura anche attraverso un puntatore.

```
struct studente s, *p;
```

```
...
```

```
p=&s;
```

```
p->nome = "Marco";
```

| | matricola | nome | cognome | media | corso |
|---|-----------|-------|---------|-------|-------|
| s | 2738 | Marco | Rossi | 25.8 | B |

↑
p

Copia dei valori di una struttura

E' possibile effettuare la copia diretta dei valori contenuti in una struttura, dentro una seconda struttura, del medesimo tipo, anche usando i puntatori.

```
struct nome_struttura{
    tipo_variabile1 nome_variabile1;
    tipo_variabile2 nome_variabile2;
    ...; }s,t,*r;

r=&s;
t=*r;
```

Passaggio di struttura a funzione

Una struttura può essere passata per valore o per riferimento, attraverso l'uso di un puntatore.

```
fun(struct studente);
```

```
main()
```

```
{ struct studente s;
```

| matricola | nome | cognome | media | corso |
|-----------|-------|---------|-------|-------|
| 2738 | Mario | Rossi | 25.8 | B |

```
...
```

```
fun(s);}
```

```
fun(a)
```

```
{ printf("%d", a.matricola);}
```

2738

Passaggio di struttura a funzione

Una struttura può essere passata per valore o per riferimento, attraverso l'uso di un puntatore.

```
fun(struct studente *);
```

```
main()
```

```
{ struct studente s,*p;
```

```
p=&s;
```

```
fun(p);
```

```
...}
```

```
fun(struct studente *a)
```

```
{ a->nome="Marco";}
```

| matricola | nome | cognome | media | corso |
|-----------|-------|---------|-------|-------|
| 2738 | Marco | Rossi | 25.8 | B |

↑ p ↑ a

Passaggio di struttura a funzione

Un vettore di struttura sarà sempre passato per riferimento.

```
fun(struct studente *a)
```

```
{ a->nome = "Marco";}
```

```
main()
```

```
{ struct studente s[2];
```

```
...
```

```
fun(s);}
```

| matricola | nome | cognome | media | corso |
|-----------|-------|---------|-------|-------|
| 2738 | Marco | Rossi | 25.8 | B |
| 6259 | Piero | Bianchi | 28 | A |

↑ a

Passaggio di struttura a funzione

Un vettore di struttura sarà sempre passato per riferimento.

```
fun(struct studente *a)
```

```
{ a++;
```

```
a->nome = "Marco";}
```

```
main()
```

```
{ struct studente s[2];
```

```
...
```

```
fun(s);}
```

| matricola | nome | cognome | media | corso |
|-----------|-------|---------|-------|-------|
| 2738 | Mario | Rossi | 25.8 | B |
| 6259 | Marco | Bianchi | 28 | A |

↑ a