I RECORD



Le strutture o record

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

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

Dichiarazione del tipo record

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

Dichiarazione del tipo record

```
struct studente{
```

```
int matricola;
string nome;
string cognome;
float media;
char corso;};
```

matricola	nome	cognome	media	corso
Intero	Stringa	Stringa	Float	Char

Visibilità del tipo record

```
#include < ... >
struct ...
            ← Ambiente globale.
                  Il tipo struct è visibile ovunque nel
                  programma.
main()
{struct ...
...}
                   Ambiente locale della funzione.
                  Il tipo struct è visibile nella funzione.
fun()
{struct ...
```

Dichiarazione di variabile di tipo record

```
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 record

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

E' possibile assegnare i valori in modo diretto:

Assegnazione valori a record

```
int matricola;
struct studente{
                   string nome;
                   string cognome;
                   float media;
                   char corso; \s;
s.matricola=1234;
s.nome="Mario";
                               L'ordine di assegnazione
s.cognome="Rossi";
                               dei valori ai campi è
s.media=28.5;
                               indifferente.
s.corso='B';
```

Assegnazione valori a record

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

Visualizzazione valori di un record

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;
cout << s.nome_variabile1;</pre>
cout << s.nome variabile2;
```

Copia dei valori di un record

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;
```

Vettori di record

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 record

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

Visualizzazione valori di vettori di record

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

```
for(i=0;i<N:i++)
     {cout << v[i].matricola;
     cout << v[i].nome;
     cout << v[i].cognome;
     cout << v[i].media;
     cout << v[i].corso;}</pre>
```

Passaggio di record a funzione

Un record può essere passato per valore o per riferimento, attraverso l'uso dell'operatore &.

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

matricola nome cognome media corso

s 2738 Mario Rossi 25.8 B
```

```
main()
{struct studente s;
...
fun(s);}
```

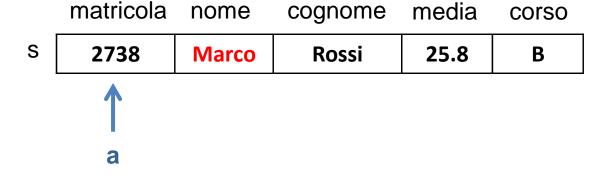
2738

Passaggio di record a funzione

Un record può essere passata per valore o per riferimento, attraverso l'uso dell'operatore &.

```
fun(struct studente &a)
{a.nome="Marco";}
```

```
main()
{struct studente s;
fun(s);
...}
```



Passaggio di record a funzione

Un vettore di record sarà sempre passato per riferimento.

```
fun(struct studente a[2])
{ a[0].nome = "Marco";}
                                 natricola
                                         nome
                                                 cognome
                                                           media
                                                                   corso
                            S
                                 2738
                                         Marco
                                                   Rossi
                                                            25.8
                                                                     В
main()
                                 6259
                                          Piero
                                                  Bianchi
                                                             28
                                                                    Α
{struct studente s[2];
```

• • •

fun(s);}

Passaggio di struttura a funzione

Un vettore di record sarà sempre passato per riferimento.

```
fun(struct studente a[2])
{ a[1].nome = "Marco";}
                                natricola
                                         nome
                                                 cognome
                                                           media
                                                                   corso
                            S
                                 2738
                                                   Rossi
                                                            25.8
                                         Mario
                                                                    В
main()
                                 6259
                                                  Bianchi
                                         Marco
                                                             28
                                                                    Α
{struct studente s[2];
```

fun(s);}