

# RIPASSO C

## Parte 3

(Slides cortesemente concesse in uso dal Prof. Fraschini)

# Puntatori e strutture

Accedo ai campi così...

**(\*punt\_stud).nome**

```
1  #include <stdio.h>
2  #define DIM_NOME 50
3
4  int main()
5  {
6      struct studente
7      {
8          char nome[DIM_NOME];
9          char cognome[DIM_NOME];
10         int matricola;
11     };
12
13     struct studente stud;
14     struct studente *punt_stud;
15
16     punt_stud=&stud;
17
18     printf("Inserisci nome studente: \n");
19     scanf("%s",stud.nome);
20     printf("Inserisci cognome studente: \n");
21     scanf("%s",stud.cognome);
22     printf("Inserisci matricola studente: \n");
23     scanf("%d",&stud.matricola);
24
25     printf("Dati dello studente inserito:\n");
26     printf("%s %s %d\n",(*punt_stud).nome,(*punt_stud).cognome,(*punt_stud).matricola);
27
28     return 0;
29 }
```

# Puntatori e strutture

o così...

**punt\_stud->nome**

```
1  #include <stdio.h>
2  #define DIM_NOME 50
3
4  int main()
5  {
6      struct studente
7      {
8          char nome[DIM_NOME];
9          char cognome[DIM_NOME];
10         int matricola;
11     };
12
13     struct studente stud;
14     struct studente *punt_stud;
15
16     punt_stud=&stud;
17
18     printf("Inserisci nome studente: \n");
19     scanf("%s",stud.nome);
20     printf("Inserisci cognome studente: \n");
21     scanf("%s",stud.cognome);
22     printf("Inserisci matricola studente: \n");
23     scanf("%d",&stud.matricola);
24
25     printf("Dati dello studente inserito:\n");
26     printf("%s %s %d\n",punt_stud->nome,punt_stud->cognome,punt_stud->matricola);
27
28     return 0;
29 }
30 |
```

# Puntatori e strutture

o così.

**stud.nome**

```
1  #include <stdio.h>
2  #define DIM_NOME 50
3
4  int main()
5  {
6      struct studente
7      {
8          char nome[DIM_NOME];
9          char cognome[DIM_NOME];
10         int matricola;
11     };
12
13     struct studente stud;
14     struct studente *punt_stud;
15
16     punt_stud=&stud;
17
18     printf("Inserisci nome studente: \n");
19     scanf("%s",punt_stud->nome);
20     printf("Inserisci cognome studente: \n");
21     scanf("%s",punt_stud->cognome);
22     printf("Inserisci matricola studente: \n");
23     scanf("%d",&punt_stud->matricola);
24
25     printf("Dati dello studente inserito:\n");
26     printf("%s %s %d\n",stud.nome,stud.cognome,stud.matricola);
27
28     return 0;
29 }
30
```

# Liste concatenate

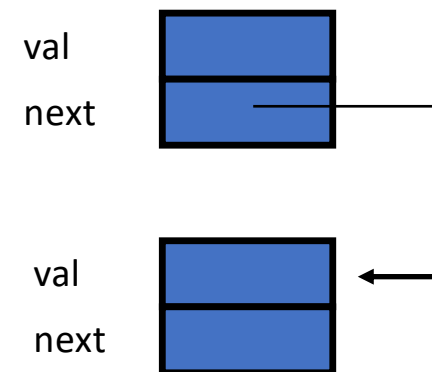
- *Puntatori a strutture e strutture con puntatori* sono di fondamentale importanza
- Permettono di creare **liste concatenate** e **alberi**

# Liste concatenate

```
1  #include <stdio.h>
2
3  int main()
4  {
5      struct valori
6      {
7          int val;
8          struct valori *next;
9      };
10
11     struct valori n1, n2;
12
13     n1.next=&n2; //il puntatore nella struttura n1 punta alla struttura n2
14
15     return 0;
16
17 }
```

**n1**

**n2**



# Liste concatenate

```
1  #include <stdio.h>
2
3  int main()
4  {
5      struct valori
6      {
7          int val;
8          struct valori *next;
9      };
10
11     struct valori n1, n2, n3;
12     int x;
13
14     n1.val=10;
15     n2.val=20;
16     n3.val=30;
17
18     n1.next=&n2;
19     n2.next=&n3;
20
21     x=n1.next->val;
22     printf("x = %d\n",x);
23     printf("%d\n",n2.next->val);
24
25     return 0;
26 }
27
```

x = 20  
30

# Liste concatenate

- Permettono di eliminare facilmente elementi da un elenco di valori
- Se volessimo **eliminare** n2 dalla lista:

**n1.next = n2.next;**

- Per **inserire** un nuovo elemento n2\_bis (tra n2 e n3):

**n2\_bis.next=n2.next;**

**n2.next=&n2\_bis;**

**Gli elementi di una lista non devono essere memorizzati in sequenza nella memoria!**



# Liste concatenate

- Importanza di un puntatore esterno che punti al **primo elemento** di una lista

**struct** valori \*punt\_lista=&n1; //con n1 già definito!

- Importanza identificatore **fine lista**
- **Puntatore nullo (puntatore NULL)** -> puntatore che non punta a nulla

**n3.next = NULL;**

# Scorri lista

```
1  #include <stdio.h>
2
3  int main()
4  {
5      struct valori
6      {
7          int val;
8          struct valori *next;
9      };
10     struct valori n1,n2,n3;
11     struct valori *punt_lista=&n1;
12
13     n1.val=10;
14     n1.next=&n2;
15
16     n2.val=20;
17     n2.next=&n3;
18
19     n3.val=30;
20     n3.next=NULL;
21
22     while(punt_lista != NULL)
23     {
24         printf("%d ",punt_lista->val);
25         punt_lista=punt_lista->next;
26     }
27
28     return 0;
29 }
30
```

10 20 30