

STRUTTURE

Persona

```
Char nome[20];
int eta;
float altezza;
```

In totale 28 byte

Creazione di una struttura Persona che ha il nome, l'età e l'altezza come parametri

Definizione di una struttura di tipo Persona

```
struct Persona {
    char nome[20];
    int eta;
    float altezza;
};
```

```
int main() {
```

```
    // Dichiarazione o creazione di una struttura di tipo Persona
    struct Persona p1;
```

```
    // Utilizzo di una variabile struttura NON PUNTATA
```

```
    p1.nome = "Mario";
    p1.eta = 23;
    p1.altezza = 1.70; // Altezza in metri
```

```
}
```

Persona p1

```
nome[20];
eta;
altezza;
```

```
nome = "Mario";
eta = 23;
altezza = 1.70;
```

File: Strutture0.c

```
Struct Persona persona1;
```

Struttura di tipo Persona chiamata persona1

persona1

```
char nome[50];
int eta;
float altezza;
```

```
Struct Persona persona2;
```

Struttura di tipo Persona chiamata persona2

persona2

```
char nome[50];
int eta;
float altezza;
```

File: Strutture1.c

```
Studiante s1;
```

```
Studiante *sp;
```

```
sp = &s1;
```

```
scanf("%s", (sp->cognome));
```

Ipotizzando che l'utente inserisca il cognome "Rossi"

s1

```
char cognome[50];
int eta;
float media;
```

sp

```
"spazio per indirizzo
di memoria"
```

sp

```
0ex5A1
```

0ex5A1

s1

```
char cognome[50];
int eta;
float media;
```

sp

```
0ex5A1
```

0ex5A1

s1

```
cognome = "Rossi";
int eta;
```

File: Strutture2.c

```
typedef struct {
    int x; // Ascissa
    int y; // Ordinata
}Punto;
```

Creo un tipo struttura nuovo di nome Punto

```
int main() {
```

```
    Punto punto1;
```

```
    Punto punto2;
```

```
    scanf("%d", &(punto1.x));
```

```
    scanf("%d", &(punto1.y));
```

```
    scanf("%d", &(punto2.x));
```

```
    scanf("%d", &(punto2.y));
```

```
    ....
```

```
    distanzaDuePunti(punto1.x, punto1.y, punto2.x, punto2.y)
```

```
    ....
```

```
    distanzaDuePuntiPuntatori(&punto1, &punto2)
```

```
    ....
```

```
float distanzaDuePunti(int x1, int y1, int x2, int y2) {
    x1 = 5;
    y1 = 6;
    x2 = 8;
    y2 = 4;
}
```

```
float distanzaDuePuntiPuntatori(Punto *p1, Punto *p2) {
```

```
p1 = 0ex5A1
```

```
p2 = 0ex5C1
```

Memoria del main()

```
int main() {
```

```
    Punto* punto1;
```

```
    Punto* punto2;
```

```
// Alloco la memoria per le due strutture puntate
```

```
punto1 = malloc(sizeof(Punto));
```

```
punto2 = malloc(sizeof(Punto));
```

```
// Prendo in input i punti
```

```
inputPunto(punto1);
```

Memoria della funzione inputPunto()

```
void inputPunto(Punto * punto)
```

punto

```
0ex5A1
```

Memoria del main()

```
0ex5A1
```

```
x = 5
y = 6
```

Ipotizziamo che l'utente abbia inserito 5 per il valore della x e 6 per il valore della y. Verranno quindi modificate le variabili nel main()