

Programmare in C

Oro o vecchiume?

Emanuele Moro dice vecchiume

Tipi di dato base

I tipi di dato in C

- Interi:
 - A. **int**: per i numeri interi con segno (positivo o negativo) 32 bit
 - B. **short int**: int, ma a 16 bit
 - C. **long int**: int, ma a 64 bit
 - D. **unsigned int**, numeri interi positivi
 - E. **signed int**, numeri interi negativi
- Virgola mobile:
 - F. **float**: numeri reali a virgola mobile rappresentabili con 32 bit
 - G. **double**: numeri reali a doppia precisione (64 bit)
- Caratteri:
 - H. **char**: un singolo carattere US-ASCII

Funzioni

Funzioni nella programmazione

Un paradigma di organizzazione del codice

La suddivisione del programma in funzioni porta notevoli benefici:

- Permettono di dare più **chiarezza** e **leggibilità** al codice, quindi;
- offrono anche una migliore **collaudabilità** e sono facili da **riutilizzare**.
- Permettono di usare la logica di scomposizione dei problemi.

Le funzioni in C

La sintassi

```
tipo_restituito nome_funzione(dichiarazione_argomenti)
{
    dichiarazioni ed istruzioni
}
```

- Se la funzione non restituisce nulla il `tipo_restituito` sarà `void`

**Chiamare una funzione sincrona,
significa fare jump sulla prima istruzione**

Variabili e Scope

Scope

La vita di una variabile

```
1  #include <stdio.h>
2
3  int main() {
4      int x = 10;
5      if (x == 10) {
6          int y = 20;
7          printf("x = %d, y = %d\n", x, y);
8      }
9      printf("%d", y);
10     return 0;
11 }
```

Scope

La vita di una variabile

```
1  #include <stdio.h>
2
3  int globalVar = 100; // Variabile globale
4
5  void printGlobal() {
6      printf("Global Variable: %d\n", globalVar);
7  }
8
9  int main() {
10     printf("Inside main: %d\n", globalVar);
11     printGlobal();
12     return 0;
13 }
```

Scope

La vita di una variabile

```
1  #include <stdio.h>
2
3  void somma(int a, int b) {
4      int risultato = a + b;
5      printf("Somma: %d\n", risultato);
6  }
7
8  int main() {
9      somma(5, 7);
0      printf("%d", a);
1      return 0;
2  }
```

Input e Output

printf("hello, world!")

La funzione printf di stdio.h è il principale **standard output** in c

Printf

```
printf("La temperatura è di %f°C", temperatura)
```

In questo caso `temperatura` è una variabile, ponendo che il contenuto della stessa sia 8.6, l'output risultante sarà:

La temperatura è di 8.6°C

Printf

```
printf("Sono le %d:%d del %d/%d/%d", h, m, d, mm, y)
```

Nel caso ci siano più variabili, l'ordine è importante ad esempio questo sarebbe l'output:

```
Sono le 12:34 del 6/03/2022 :)
```

scanf("%d:%d", &h, &m)

La funzione scanf di stdio.h è il principale **standard input** in c

Non c'è solo lo standard input...

Riga di comando

Non è standard input

- Viene utilizzato più spesso per inserire file, o directory all'avvio del processo (istanza del tuo programma).
- Ma puoi inserire qualsiasi tipo di valore.
- Ricordati però che ogni cosa che viene dalla riga di comando è stringa e va convertita se necessario.

```
#include <stdio.h>
```

```
int main(int argc, char *argv[]) {
```

```
...
```

```
}
```

`#include <stdio.h>` Questa * (star) indica che è un puntatore

```
int main(int argc, char *argv[]) {  
    ...  
}
```

Contatore int degli argomenti passati

Puntatore ad un array di caratteri (stringa)

Riga di comando

Esempio di funzionamento

- Poniamo il caso di avviare sul nostro computer windows, un programma che fornisce statistiche sulle temperature registrate ogni 10 minuti per 12 mesi da 2 sensori. Si mostrerà come da esempio.

tempstat	".\sensor1.txt"	".\sensor2.txt"
0	1	2

argc = 3

argv[0] = tempstat

argv[1] = .\sensor1.txt

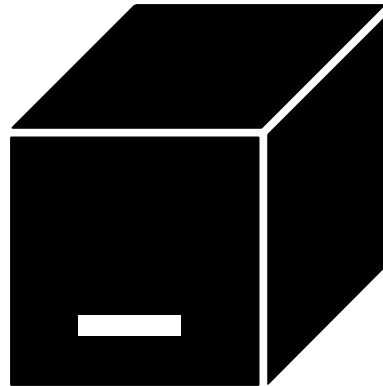
argv[2] = .\sensor2.txt

Vettori

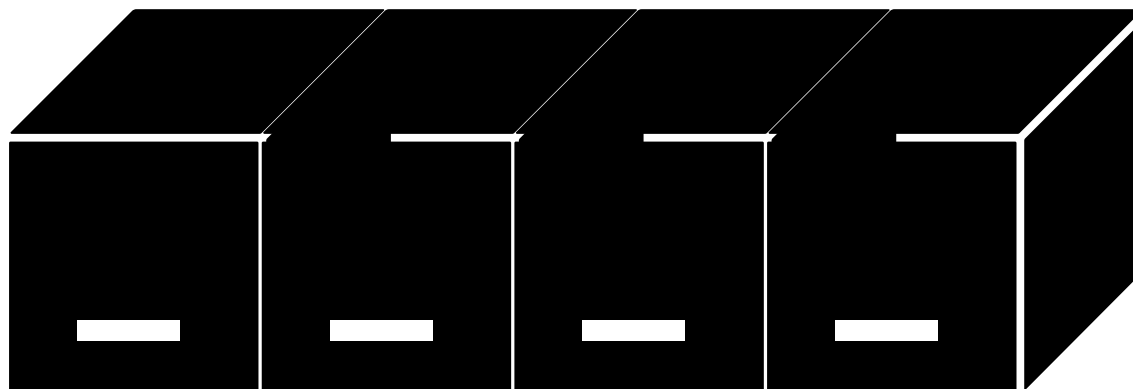
Questa è una variabile.



Nome

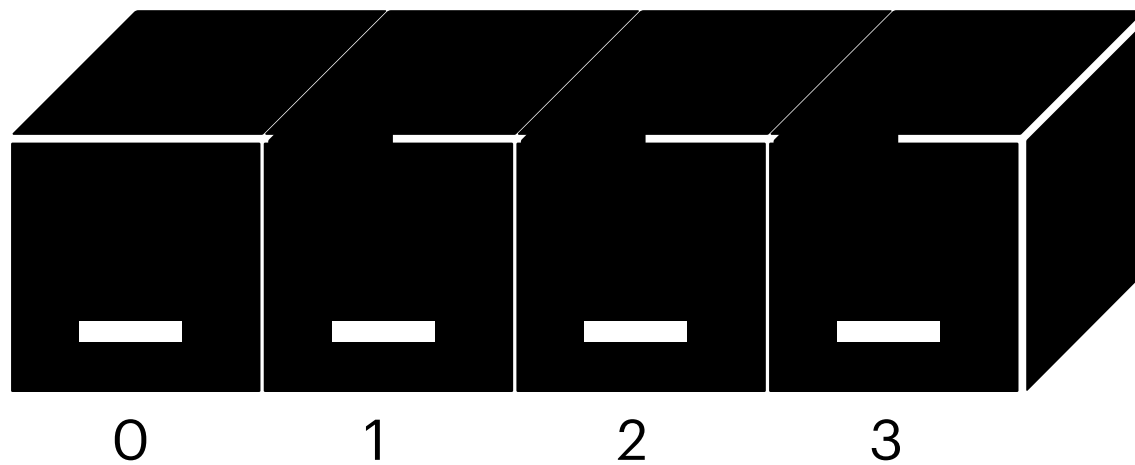


Nome



Questo è un array

Nome



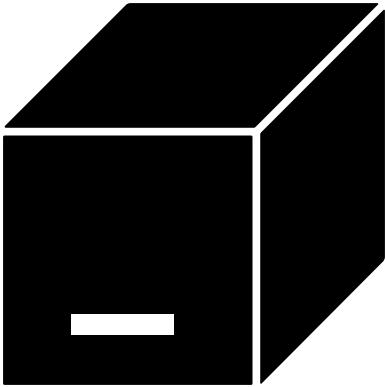
Ha un sistema posizionale

Puntatori
Que dolor

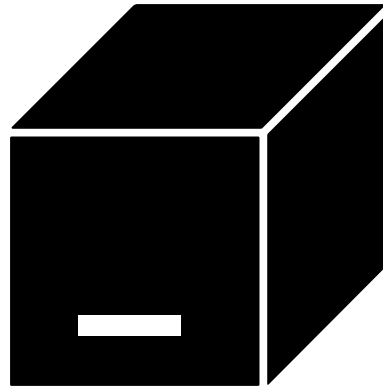
Questa è una variabile.



Nome



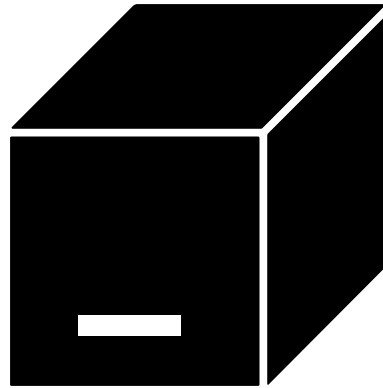
Nome



Questa no

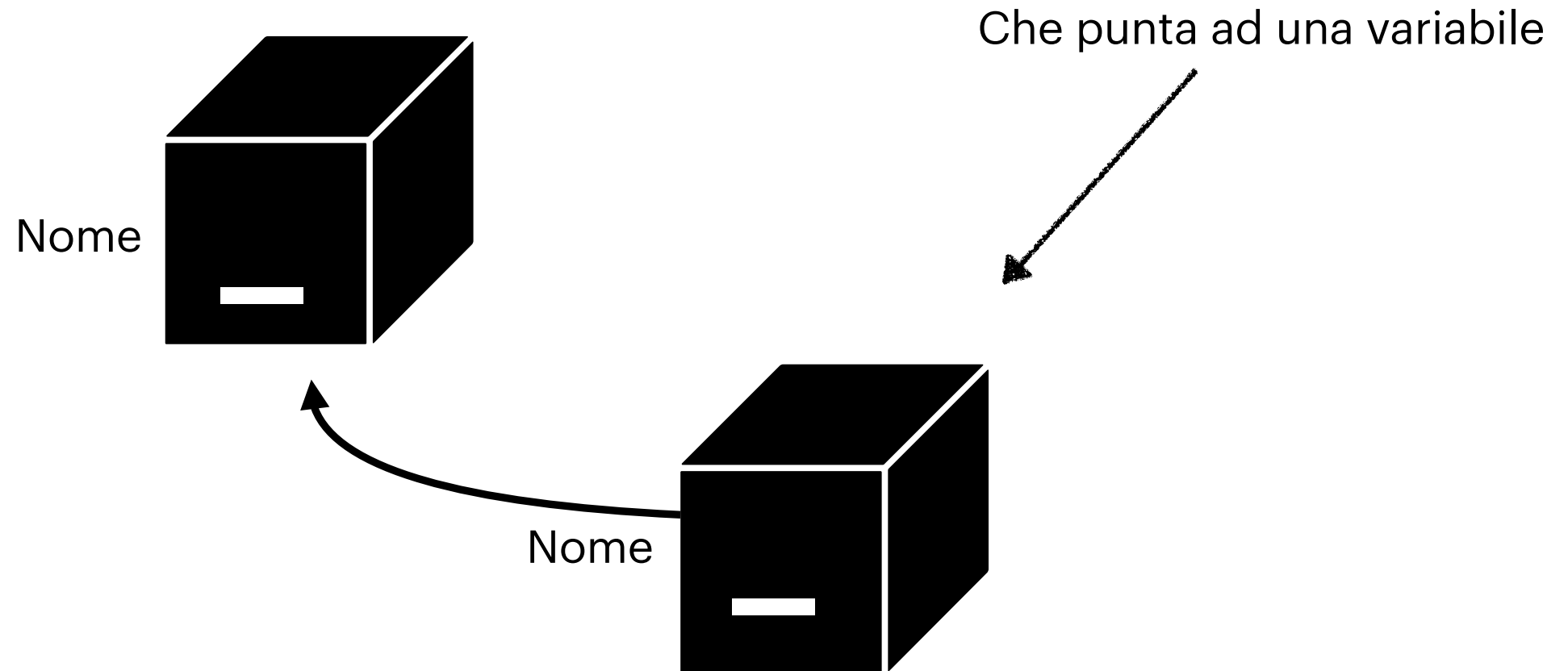


Nome

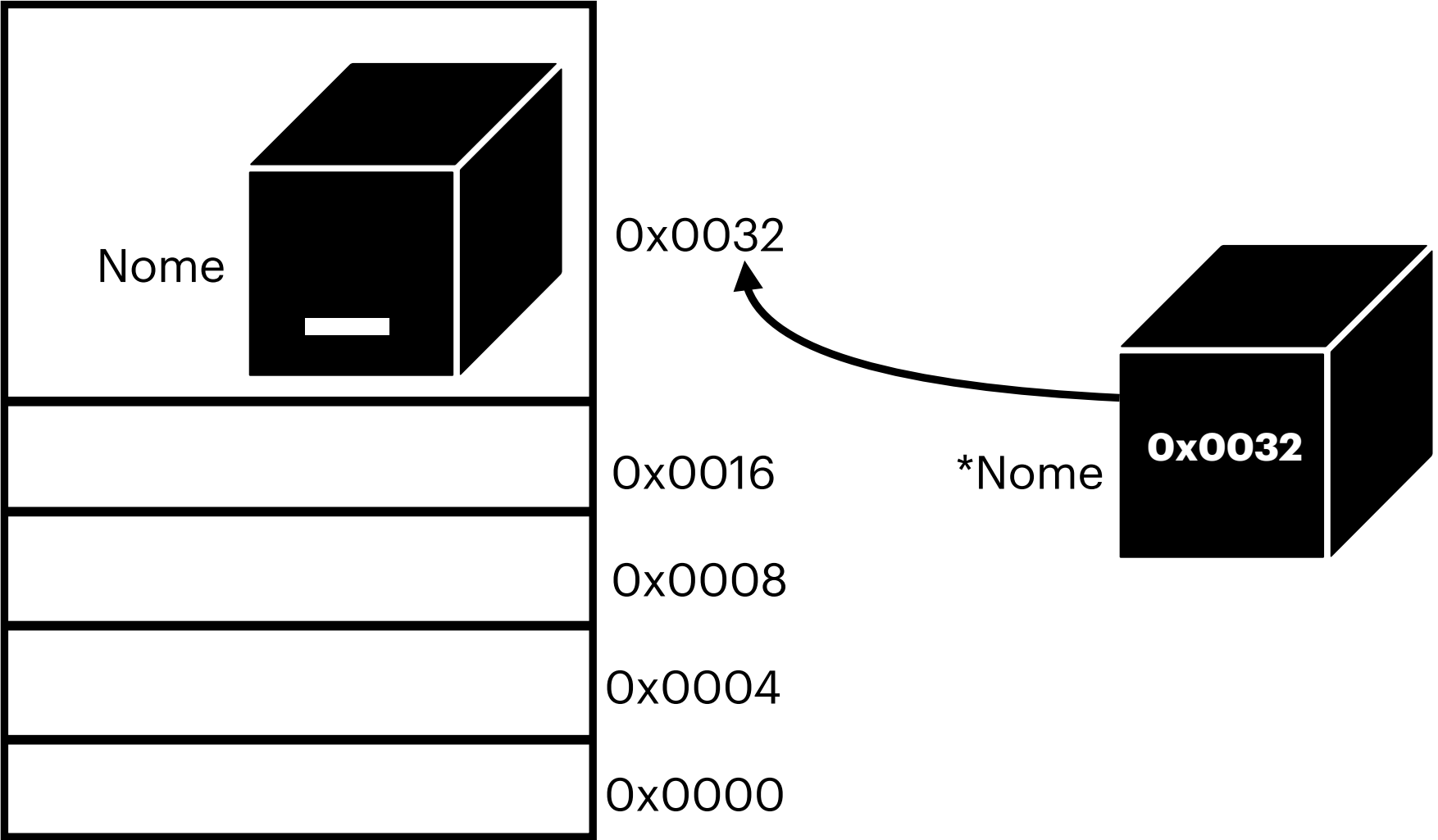


È un puntatore





In sostanza è una variabile che ha come valore l'indirizzo di memoria di una variabile



ESERCIZIO

Gestione di un Registro Studenti con Input da File