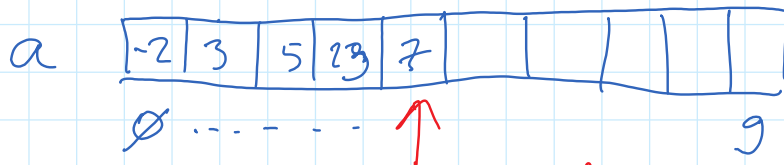
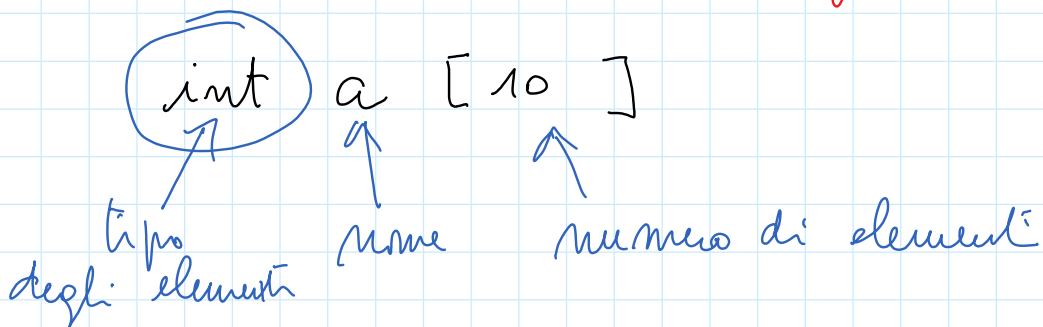


# ARRAY

mercoledì 27 settembre 2017 08:55

Sequenza di elementi, tutti dello stesso tipo, individuabili attraverso un indice che va da 0 alla lunghezza - 1 dell'array.

Dichiarazione di un array



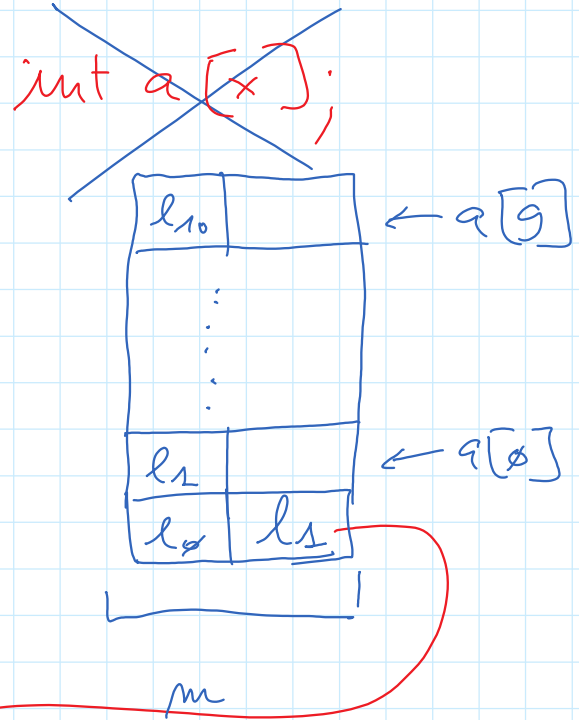
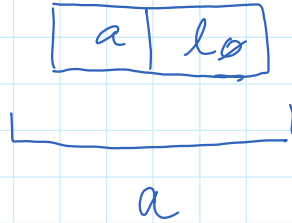
Ciascun elemento è una variabile

Notazione

`a[i]` l'elemento di indice `i` dell'array `a`

~~int~~ `int a[100];`  
↑  
costante

è vista come  
una variabile  
puntatore

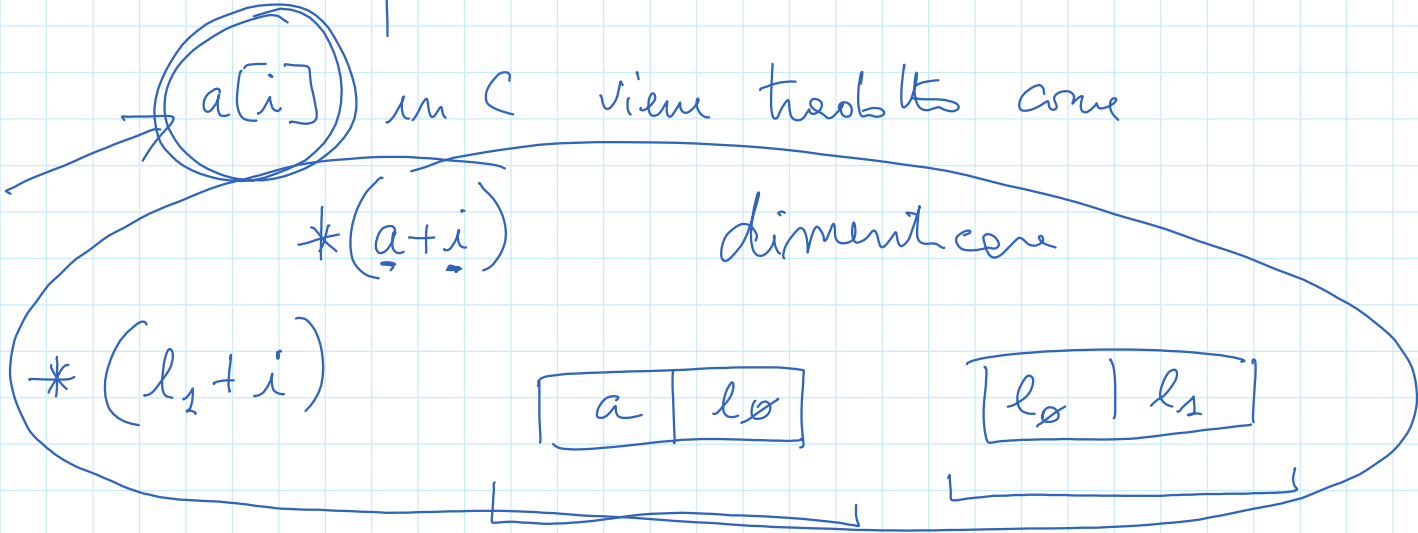


l'indirizzo di memoria  
della variabile "primo elemento  
dell'array"

$$a[3] = a[3] + 1;$$

↑  
località dell'  
elemento di  $a$   
di indice 3

↑  
valore dell'elemento di  $a$   
di indice 3



```
} int a[100];
```

```
for (int i = 0; i < 100; i++)
```

```
{ a[i] = 0; }
```

```
{ int x = 0;  
  a[i] = x;  
}
```

input &

```

int a[100];
int el = 10;
int i = 0;
lettura dell'array

```

$\wedge$ :  $a \wedge b$  è vero sse  
 $a$  e  $b$  sono veri.

(prendere 100 elementi  
 interi dall'esterno  
 (da tastiera, da file)  
 e assegnare i valori  
 agli elementi dell'array)

```

while (i < 100 &&
       a[i] != el)
  i++;

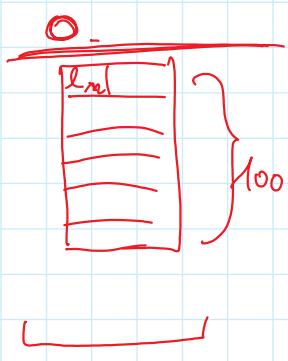
```

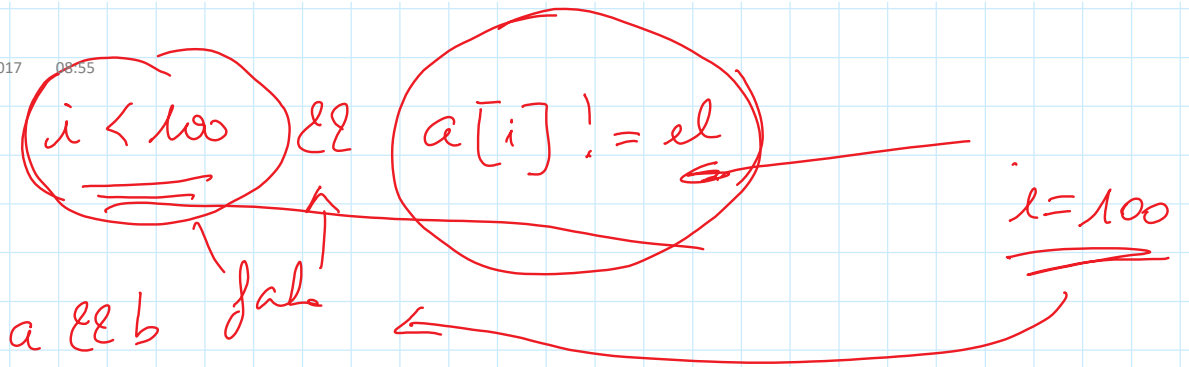
```

if (i < 100)

```

a[100]





- esecuzione:
- valuta il valore di  $a$
  - valuta il valore di  $b$
  - se entrambi sono true il risultato è true, altrimenti false.

Se entrambi gli argomenti di  $\&\&$  vengono valutati, nel caso in cui  $el$  non compare nell'array la condizione dà errore (concettuale)

Esiste un operatore che si chiama

mercoledì 27 settembre 2017 08:55

And condizionale (and)

$a \text{ and } b = \text{if } a \text{ then } b$   
↑ ↑  
else false

es and condizionale

$(i < 100 \text{ \&\amp; } a[i] \neq el)$   
↑  
 $i = 100$  non viene valutata

$(a[i] \neq el \text{ \&\amp; } i < 100)$   
in C può dare errore

# Variabile logica

mercoledì 27 settembre 2017 08:55

il cui valore è un elemento di verità (true oppure false)

~~boolean x = true;~~

$\emptyset \rightarrow$  false  
 $\overline{1} \rightarrow$  true

(tutti i valori interi diversi da  $\emptyset$  rappresentano il valore true)



```
int a[100];
int el = 10;
int i = 0;
int trovato = 0;
while (i < 100 && ! trovato)
    if (a[i] == el) trovato = 1;
    else i++;
if (trovato)
    (trovato == 1)
```

not

non ho trovato  
l'elemento

trovato == 0

funclé è  
true il not  
di trovato

```
while (i < 100 && ! trovato)
{
    if (a[i] == el) trovato = 1;
        i++;
}
```

trovato = 0       $\Rightarrow$       ! trovato = 1  
"      = 1       $\Rightarrow$       ! " = ~~0~~

# Variabili puntatore

mercoledì 27 settembre 2017 08:55

```
void swap (int * m, int * n)
```

```
{
    int temp = *m;

```

```
    *m = *n;
    *n = temp;
}
```

prendi il valore  
alla locazione che  
è valore di n

prendi il valore  
che c'è in memoria  
alla locazione che  
è valore di m

metti il valore nelle  
locazioni che  
manti

```
{
    int x = 10;
    int y = 20;

```

```
    swap (&x, &y)
```

temp	l4
------	----

m	l3
n	l2

y	l1
x	l0

swap	def
------	-----

a

l4	10
----	----

l3	l1
l2	l0

l1	<del>20</del> 10
l0	<del>10</del> 20

m

Operazioni di input/output  
lettura / scrittura  
ingresso / uscite } syn

procedure predefinite

librerie { raccolte di procedure  
e funzioni che  
possono essere incluse  
nel nostro programma

STDIO standard input/output

#include <stdio.h>

# scanf

legge valori dall'esterno (testiere, file...)  
e associa a variabili del programma

anche l'input

formato

(secondo il formato  
che tu dici)

~~345~~

scanf (" %d ", &x)

variabili  
a cui  
assegnare il  
valore

↑  
stringa di caratteri:  
sequenza

`printf("%d", x);`  
espansione

`printf("il valore e': %d ma non basta", x);`

il valore e': 10 ma non basta

procedure che legge un array

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

```
void read (int v [])
```

```
{  
  for (int i=0; i < 10; i++)  
    scanf ("%d", &a[i])  
}
```

```
main ()
```

```
{ int a [10];
```

```
  int b [15];
```

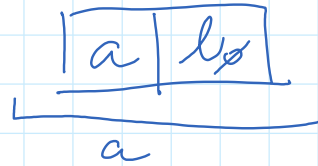
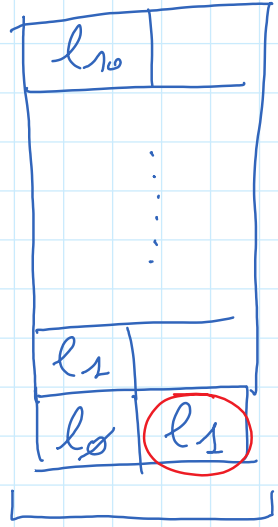
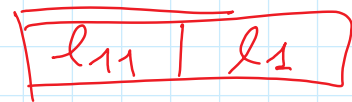
```
  read (a);
```

```
  read (b);
```

```
main()  
{ int a [10];
```

```
read(a);
```

void read (int v[])





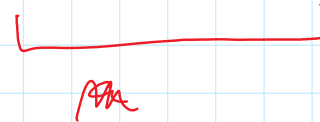
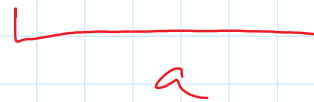
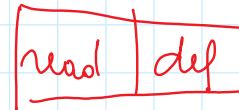
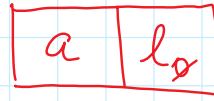
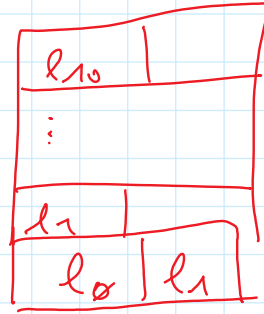
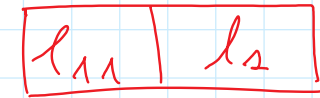
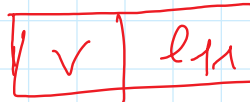
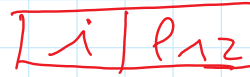
void read (int v [])

mercoledì 27 settembre 2017 08:55

```
{ for (int i = 0; i < 10; i++)  
    scanf ("%d", &a[i]);  
}
```

main ()

```
{ int a [10]  
  read (a);  
}
```

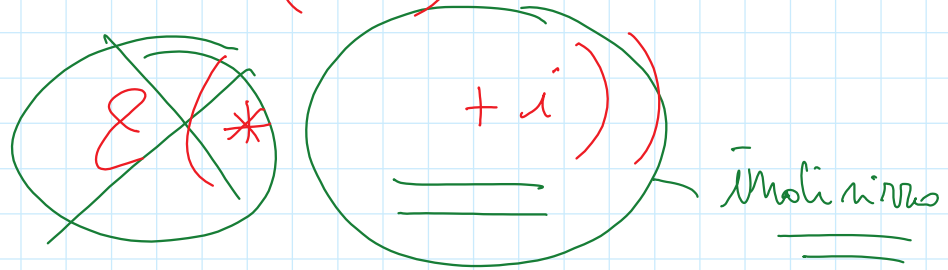


$\text{scanf}("%d", \&v[i])$

$v[i]$   
 $\ast(v+i)$

$v[i]$   
 $\ast(v+i)$

$\ast(v+i)$



$$v[i] = v[i] + 1$$

$$\ast(v+i) = \ast(v+i) + 1$$