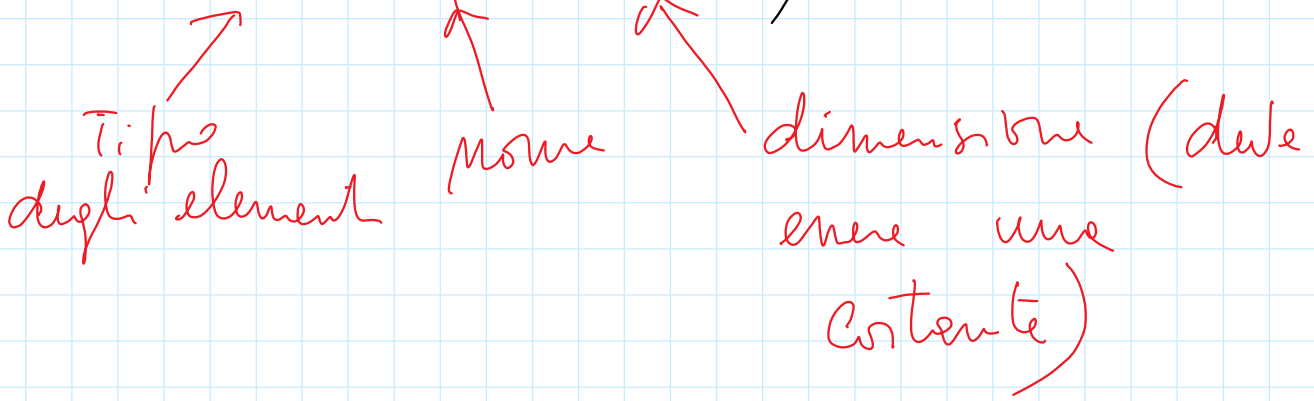




# Di dichiarare un array

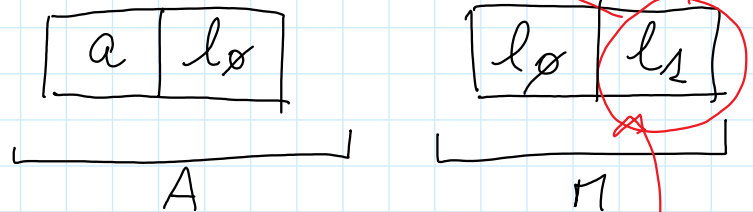
```
int a [10];
```



```
int a [5];
a[0] = 0;
```

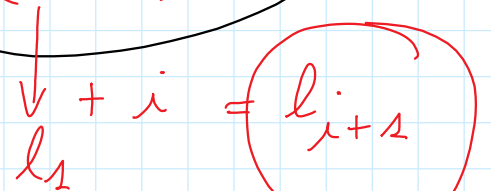
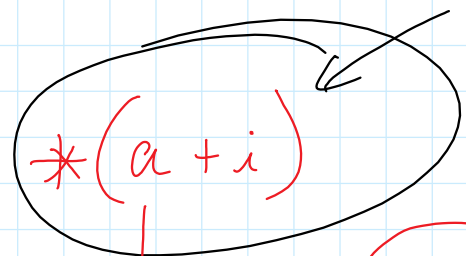
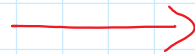
$l_5$	?
$l_4$	?
$l_3$	?
$l_2$	?
$l_1$	?

l'indirizzo del primo dei 5 element che compongono a



a è una variabile puntatore particolare (il suo valore non può essere modificato)

a[i]



$$v^T \wedge l_1 = x_{\lambda+1}$$

$l_1 + 0 = l_1$  lo course del primo elemento di  $a$

$l_1 + 1 = l_2$  " " secondo " di  $a$

e così via

```

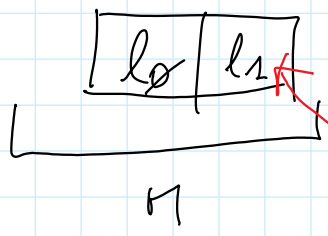
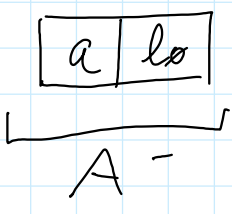
} int a[5];

```

$a[0] = 0;$   
 $a[1] = a[0] + 1$

$* (a + 0) = 0;$   
 $* (a + 1) = * (a + 0) + 1;$   
 Induzione

$l_5$	?
$l_4$	?
$l_3$	?
$l_2$	?
$l_1$	?



$* a = 0$   
 ↑  
 valore di a  $l_1$

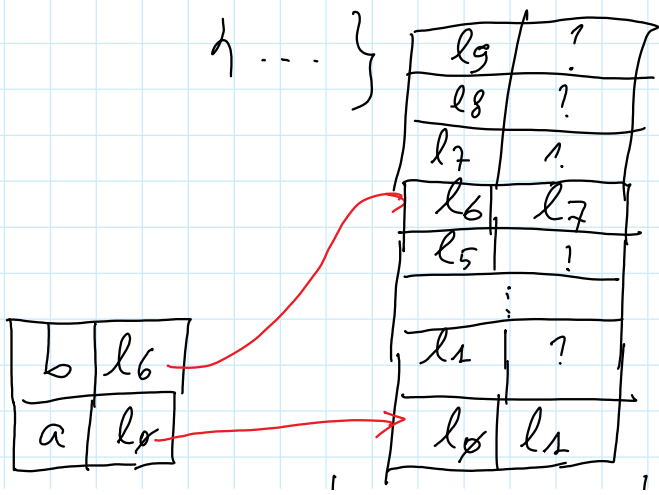
Scrivere una procedura che ordini tutti gli elementi di un array.

```

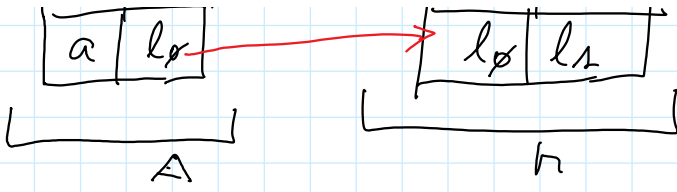
{ int a[5];
  int b[3];
  void ordine ( ... )
  { ... }

```

due array  
 tutti gli elementi  
 di un array  
 passati come  
 argomenti.



il valore  
 di a  
 che cosa  
 $ordine(a);$  ?



arrivare  $(a)$ ;  $z$ !  
arrivare  $(b)$ ;

```
{ int a[5];  
  int b[3];
```

```
void arrune (int v[], int dim)  
{ int i;  
  for (i=0; i < dim; i=i++)  
    v[i] = 0;  
}
```

~~int \*v~~

int v[]

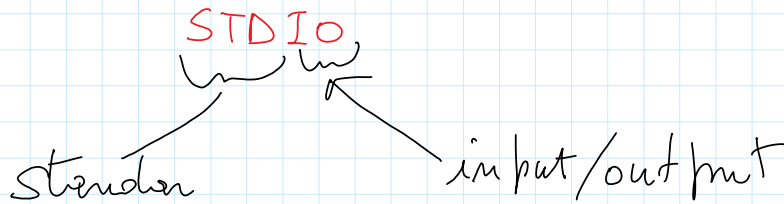
i++

&a

```
main()  
  arrune (a, 5);  
  arrune (b, 3);
```

Comunicazione di un programma con l'esterno è fatta in C attraverso procedure di input/output

Le procedure di i/o sono contenute (le più semplici) in una libreria (collezione di funzioni e procedure già definite) che si chiama



La libreria si "INCLUDE" attraverso la direttiva (che va messa in testa al programma)

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

scanf  
printf

*format* (with arrow pointing to scanf)

scanf ( , )

formato dei valori da leggere (stringa di caratteri tra doppie aprie "...")

le variabile

doppio apice "..."

le variabile  
a cui  
assegnare il  
valore letto

posso usare caratteri  
speciali che danno il formato

es: %d valore intero

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



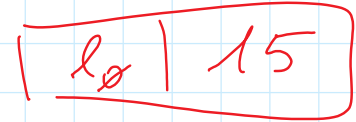
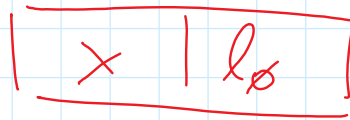
`printf("il valore di x è: %d",`



il valore  
dell'espressione

il valore di x è: 15

2° argomento



`printf("il valore di x è: %d e quello  
di y è: %d", x, y);`

funzione che fa la somma di due valori interi

```

int somma (int m, int n)
{
  return m + n;
}

```

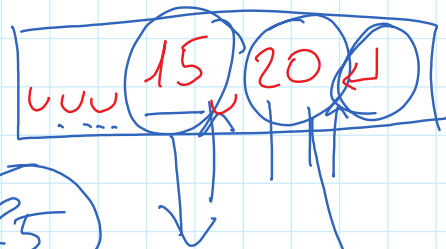
25

main()

```

int x;
int y;
scanf ("%d", &x);
scanf ("%d", &y);

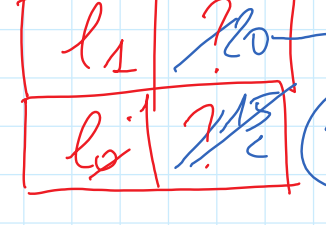
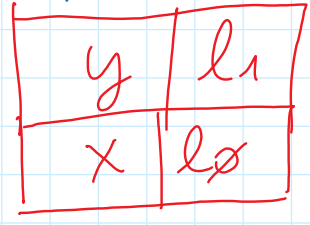
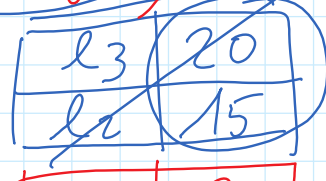
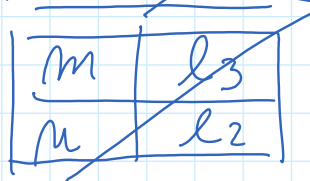
```



```

x = somma (x, y);

```



25

A

M

```

int somme (int * m, int * m)
{ return *m + *m; }

```

```

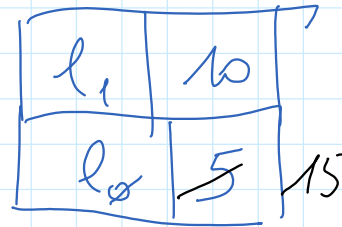
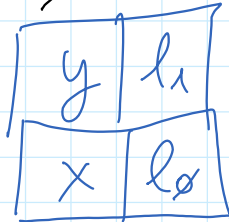
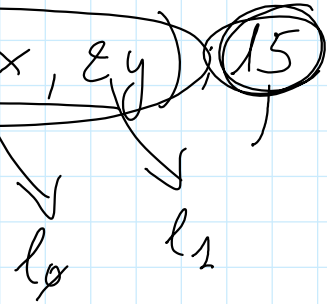
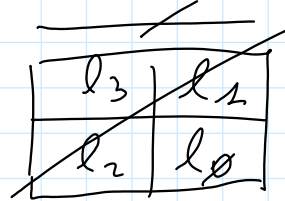
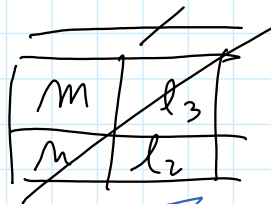
main()
{ int x = 5;
  int y = 10; }

```

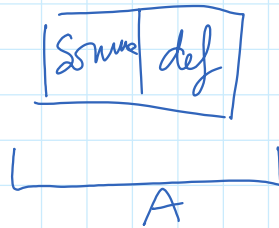
```

x = somme (&x, &y) 15

```



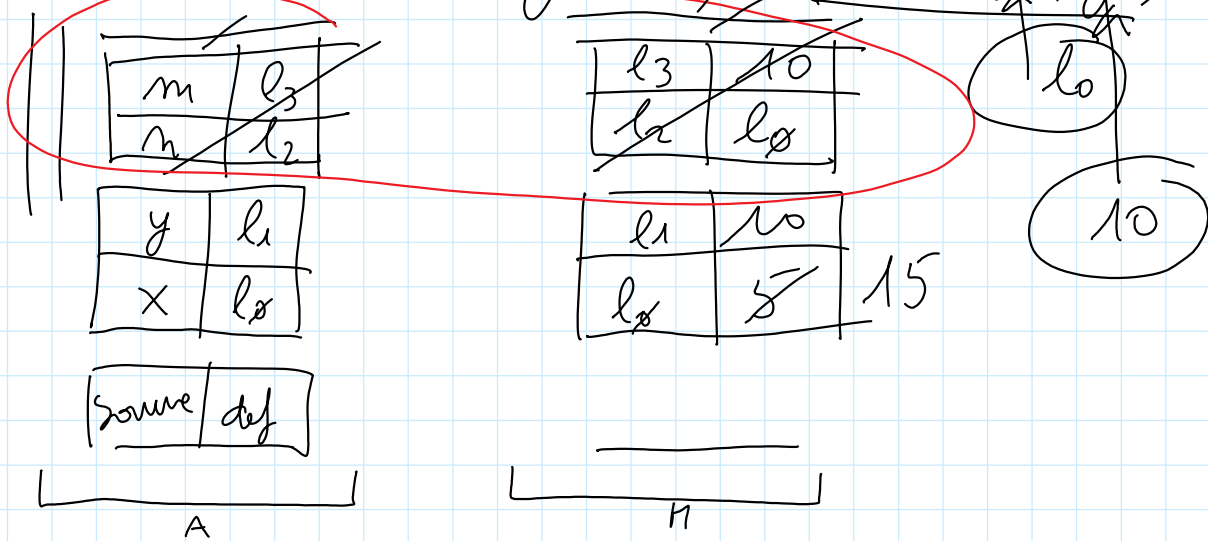
\*m val 5  
\*m val 10



Somme non deve più essere una funzione, ma una procedura che modifica il valore di una variabile con la somma del valore della variabile stessa e di un'altra variabile.

```
void somma (int *n, int m)
{
  *n = *n + m;
}
```

```
main()
{
  int x = 5; int y = 10; somma(&x, y);
}
```



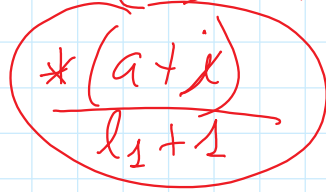
# Procedura m array

mercoledì 26 settembre 2018 11:57

```
void arraye (int a[], int dim)
```

```
{ int i;
  for (i=0; i < dim; i++)
```

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



main()

```
{ int a[3];
  arraye (a, 3);
```

