Titolo nota 24/09/2015

```
int * x;
int y;
```

il valore della variabile non è un intero ma è l'indirizzo di memoria di un valore intero

```
x = y;
```

ERRORE

Ambiente e Memoria servono per ESPLICITARE gli indirizzi di memoria (locazioni)

y	l_1
x	l_0

Ambiente

l_1	5
l_0	l_1

Memoria

non può essere un valore intero

l_1 è l'indirizzo di memoria di un valore intero (in questo caso 5)

Titolo nota 24/09/2015

In C c'è la possibilità di indicare, oltre al valore, l'indirizzo di una variabile

```
int *x;
int y = 5;
x = &y;
```

l'indirizzo della variabile y (cioè la locazione associata a y nell'ambiente)

CORRETTO

Se x ha come valore l'indirizzo di un intero (5) è possibile "arrivare" a quel valore?

```
int *z;
z = x;
```

vale l1?

y	l1
x	l0

Amb.

l1	5
l0	?

l1

Mem

Titolo nota 24/09/2015

Come "arrivare" da una variabile puntatore al valore "puntato"?

Mediante l'espressione `*x` il valore intero puntato da x, cioè il valore intero che ha come indirizzo il valore di x

```
int *x;
int y = 5;
x = &y;
```

Qual'è il valore di:

- x → l_1
- *x → 5
- &x → l_0

y	l_1
x	l_0

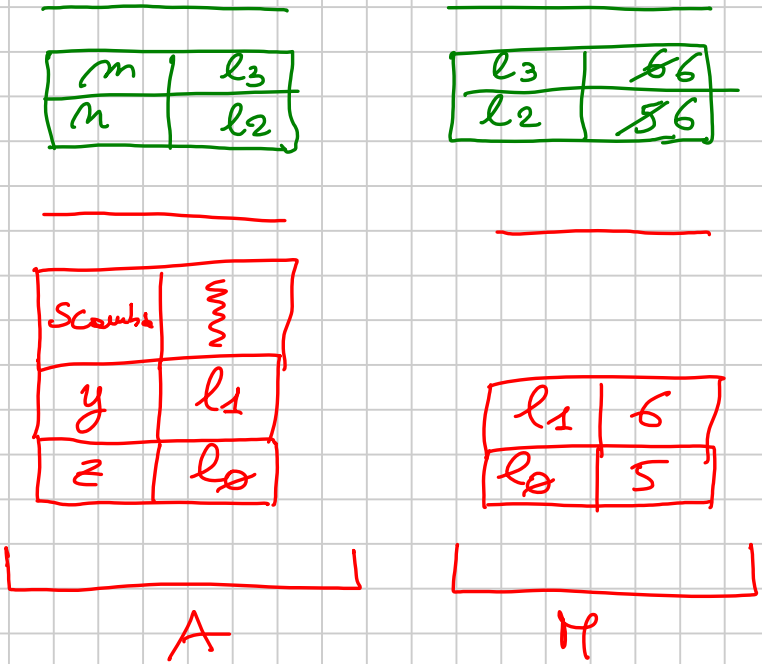
l_1	5
l_0	l_1

Titolo nota 24/09/2015

```
int z = 5;
int y = 6;
void scambia (int n, int m)
{
    n = m;
    m = n;
}
main ()
{
    scambia (z, y);
}
```

NO

5 6



Titolo nota 24/09/2015

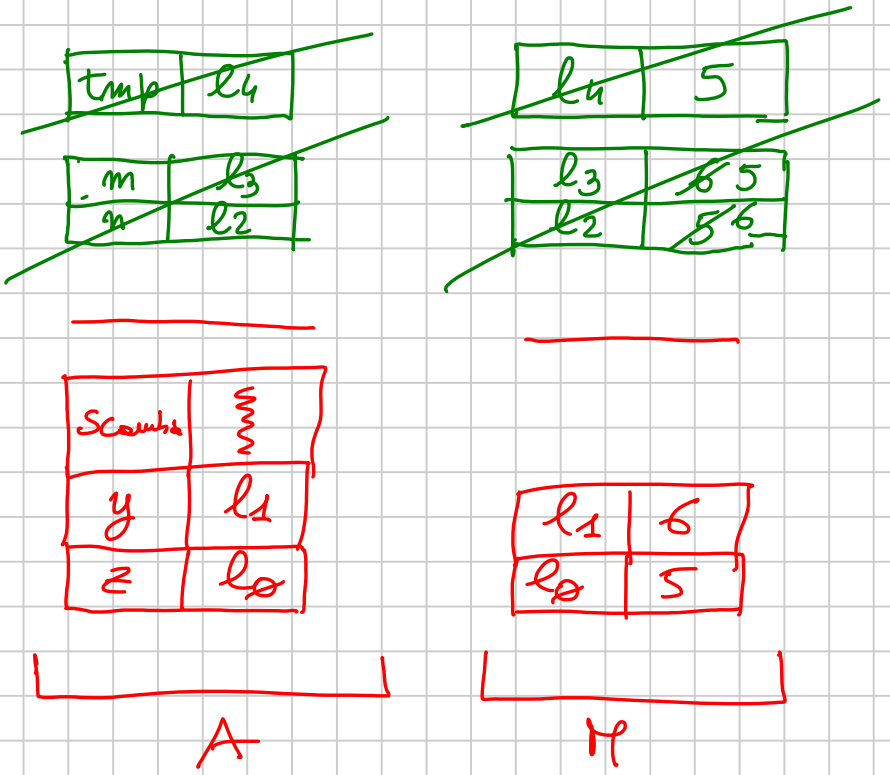
```
int z = 5;
int y = 6;

void scambia (int n, int m)
{
    int tmp = n;
    n = m;
    m = tmp;
}

main ()
{
    scambia (z, y);
}
```

5 6

PASSAGGIO DEI PARAMETRI IN C è "PER VALORE"

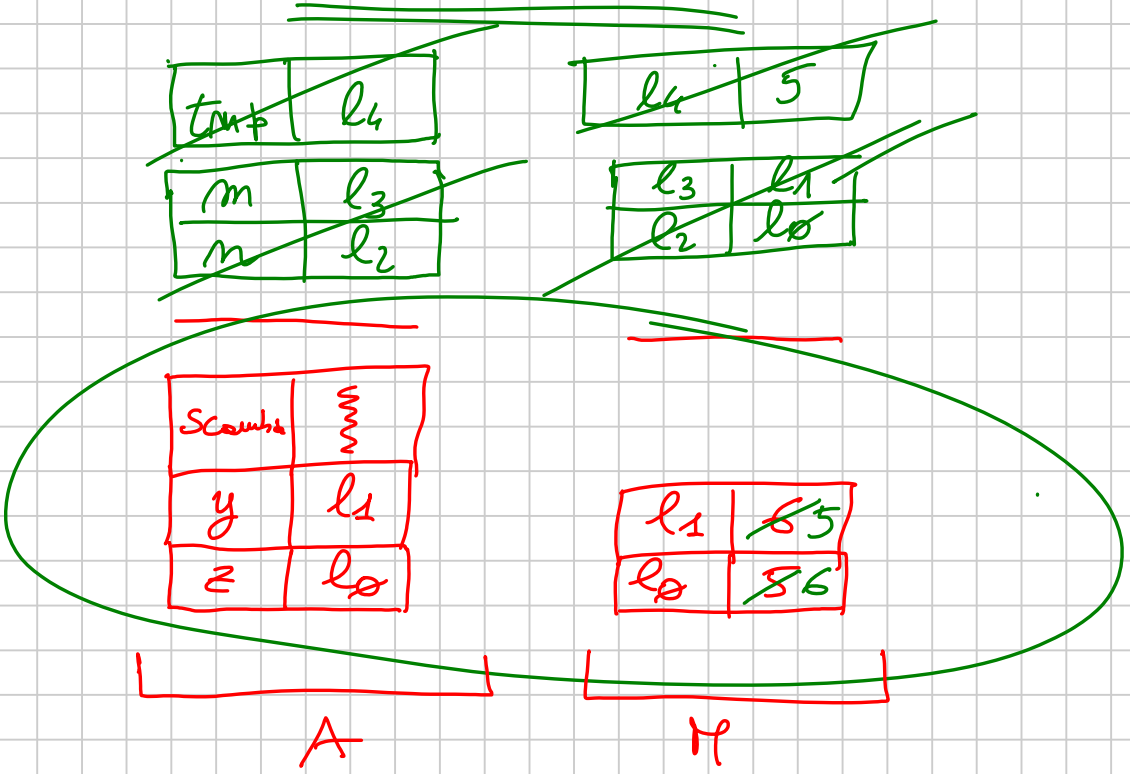


Titolo nota 24/09/2015

```
int z = 5;
int y = 6;
void scambia (int*m, int*m)
{ int tmp = *m;
  *m = *m;
  *m = tmp;
}
main ()
{
  scambia(&z,&y);
}
```

5
6
5

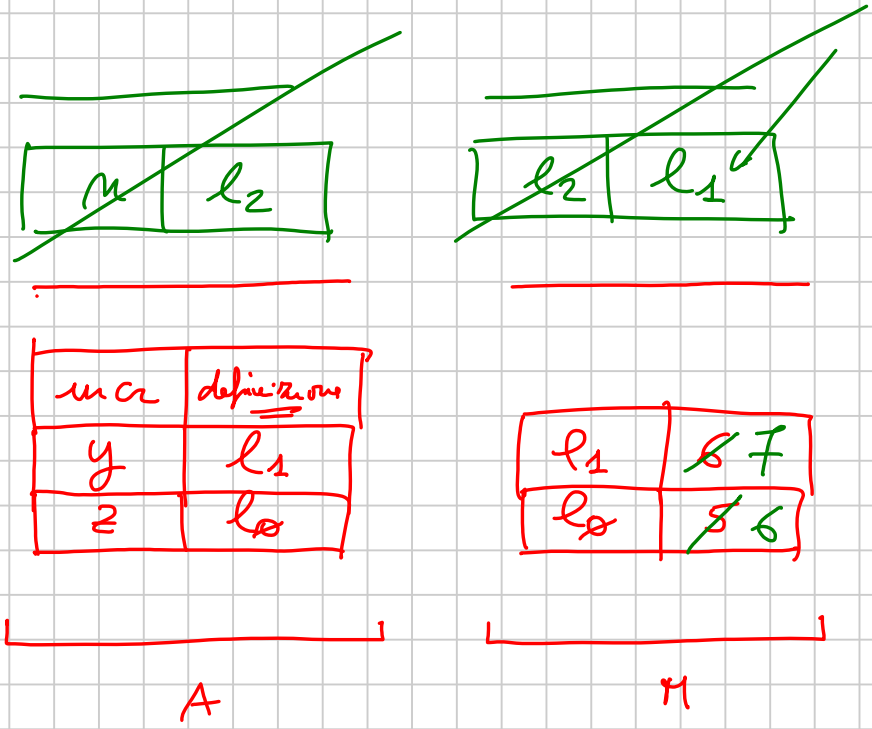
PASSAGGIO DEI PARAMETRI IN C "PER VALORE"



```
{
  int z = 5;
  int y = 6;
  void inc (int * n)
  { *n = *n + 1; }
  main ()
  {
    inc (&z);
    inc (&y);
  }
}
```

6

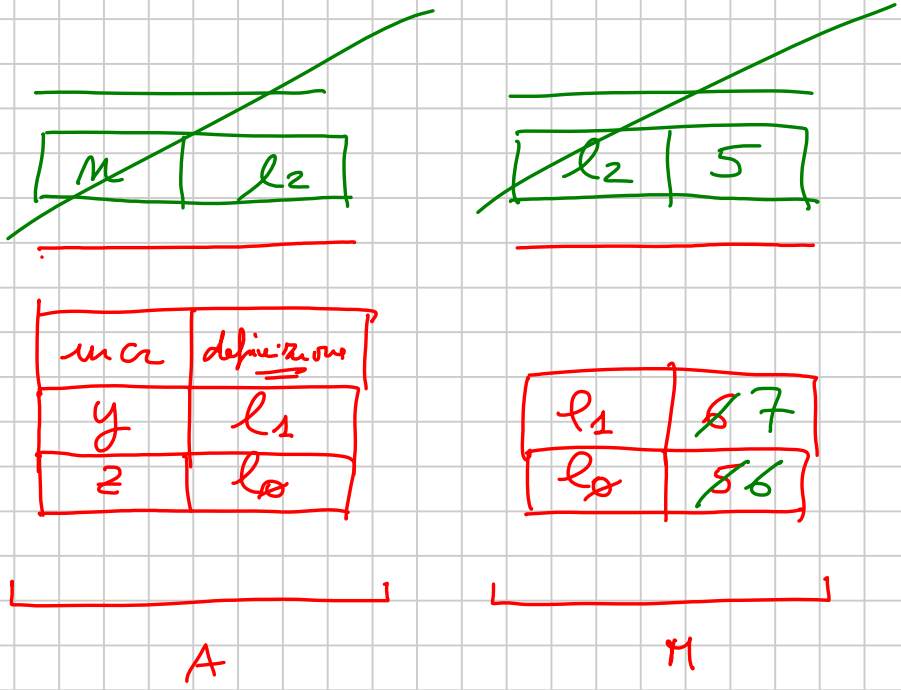
l_1



```
{
  int z = 5;
  int y = 6;
  int inc (int n)
  {return n+1;}

main ()
{
  z = inc (z);
  y = inc (y);
}
```

6



Titolo nota 24/09/2015

```
{
  int z = 5;
  int y = 6;

int incr (int n, int *m)
{
  *m = *m + 1;
  return n + 1;
}

main ()
{
  z = incr (z, &y);
}
```

7

6

5 l_1

m	l_3
n	l_2

l_3	l_1
l_2	5

incr	definizione
y	l_1
z	l_0

l_1	~~6~~ 7
l_0	~~5~~ 6

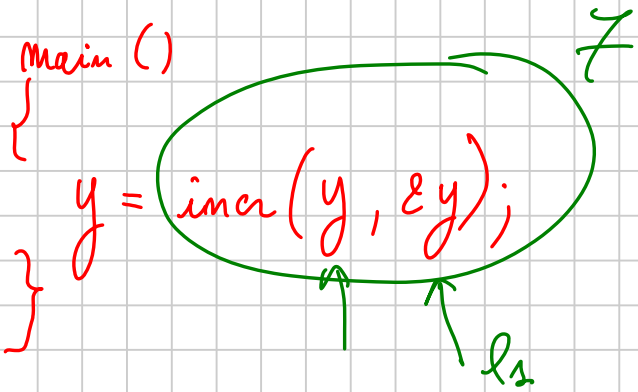
A M

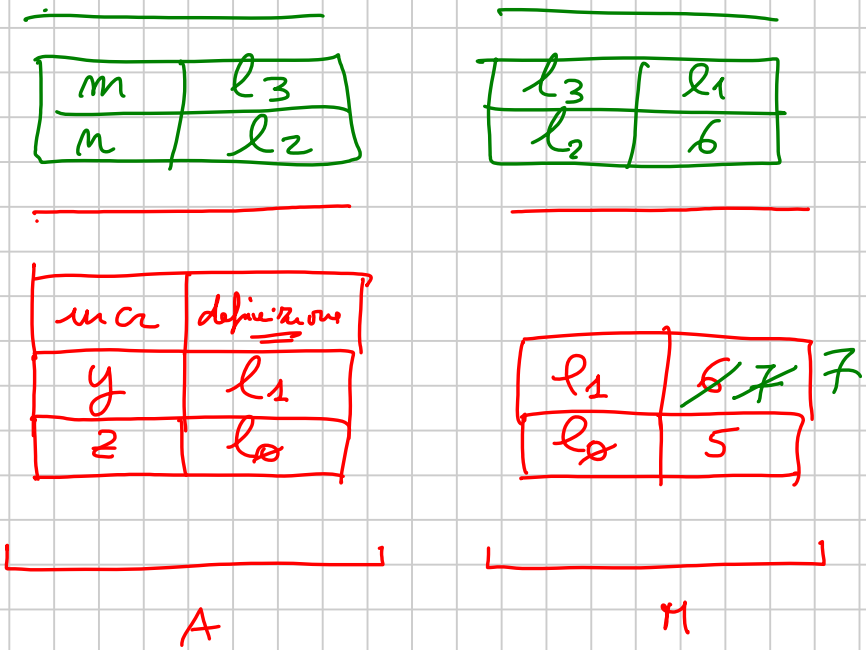
Titolo nota 24/09/2015

```
{
  int z = 5;
  int y = 6;

int incr (int m   int *m)
{
  *m = *m + 1;
  return m + 1;
}

main ()
{
  y = incr(y, &y);
}
```





```
{
  int z = 5;
  int y = 6;

int incr (int n, int *m)
{
  *m = *m + 1;
  return y + 1;
}

main ()
{
  y = incr(y, &y);
}
```

8

6 ℓ_1

variabile "globale"

non usare variabili globali
(sono poco leggibili)

m	ℓ_3
n	ℓ_2

ℓ_3	ℓ_1
ℓ_2	6

incr	definizione
y	ℓ_1
z	ℓ_0

A

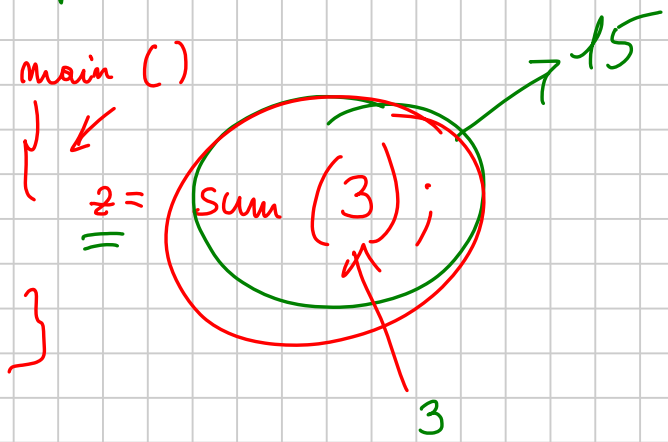
ℓ_1	~~6~~ 7 8
ℓ_0	5

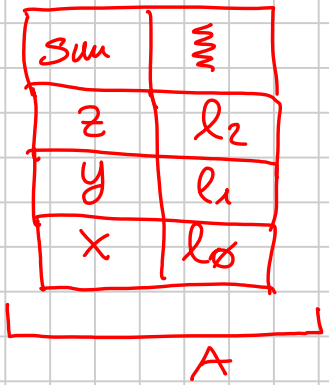
M

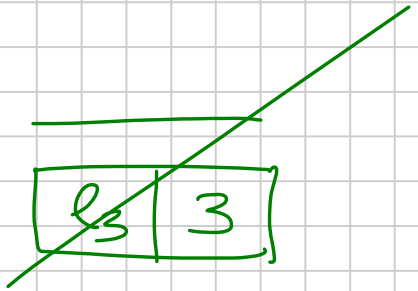
Titolo nota 24/09/2015

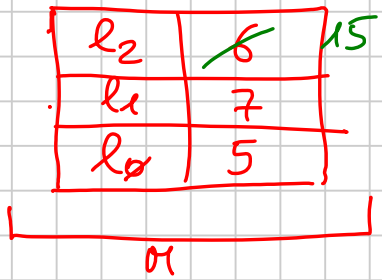
```
int x = 5;
int z = 6;
int y = 7;
int sum (int n)
{return x + y + n;}

main ()
{
  z = sum (3);
}
```









PORTATA DEGLI IDENTIFICATORI (scoping) STATICA

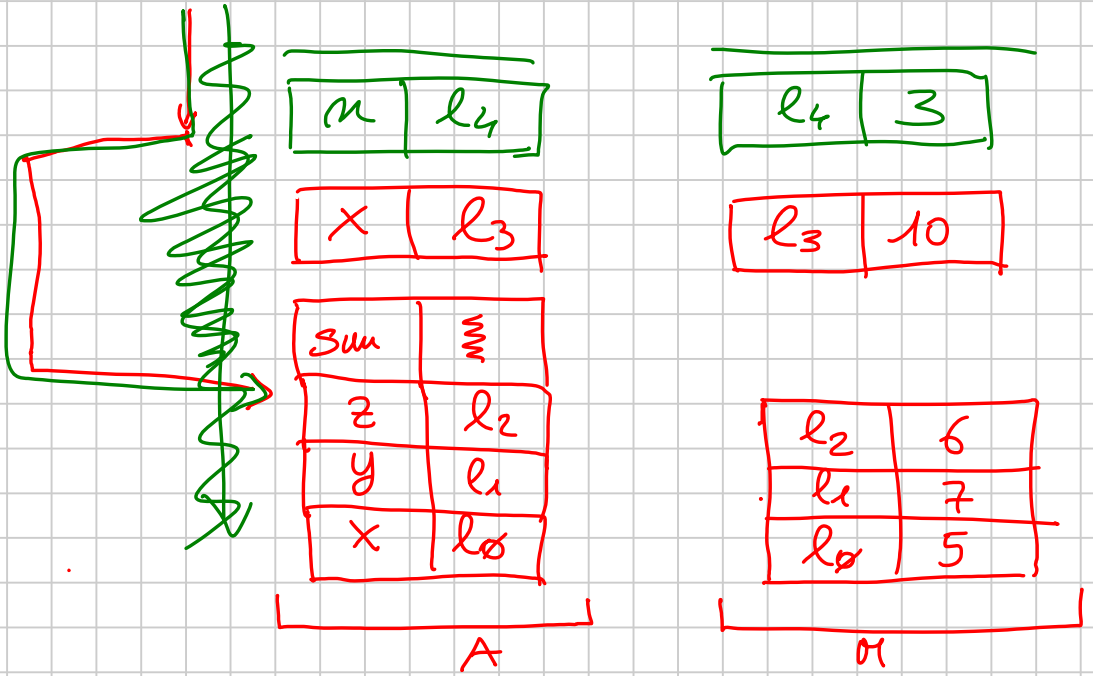
Titolo nota 24/09/2015

```
{ int x = 5;
  int z = 6;
  int y = 7;
int sum (int n)
{ return x + y + n; }

main ()
{
  int x = 10;
  z = sum (3);
}
```

x vale ~~10~~

Le variabili globali vanno cercate nell'ambiente al momento della dichiarazione della funzione

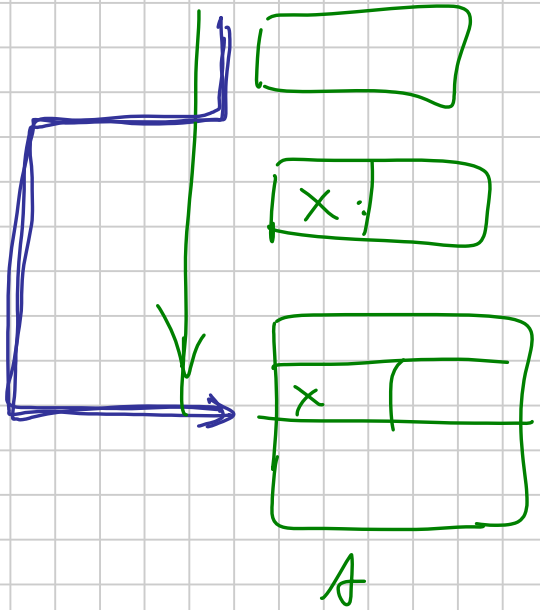


Titolo nota 24/09/2015

Linguaggi in cui le variabili globali si cercano scandendo le pile nel modo usuale (dall'alto verso il basso)

PORTATA DEGLI IDENTIFICATORI DINAMICA

catena statica



PORTATA DEGLI IDENTIFICATORI (scoping) STATICA

Titolo nota 24/09/2015

```
{ int x = 5;
  int z = 6;
  int y = 7;
  int sum (int n)
  { return n + 2; }
```

```
main ()
{
  int x = 10;
  x = sum (x);
}
```

12

10