ambiente

y	l_2
x	l_0

memoria

l_1	7
l_0	5

Il C è un linguaggio "A BLOCCHI"

{ x = x + 1; y = y + 2 }

è un unico comando

dichiarazione: int x = 5;

comando: x = 5;

BLOCCO

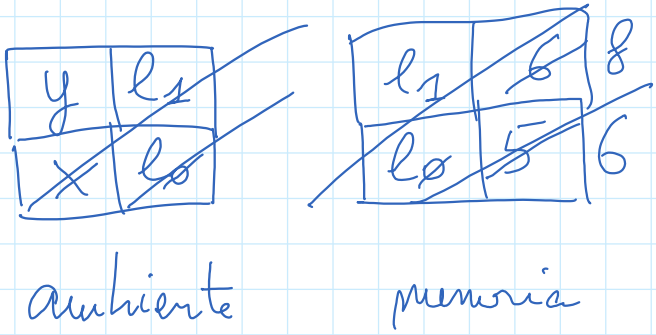
{ Sequenza di dichiarazioni
Sequenza di comandi }

COMANDO

Espressioni calcolano valori
Dichiarazioni creano lo stato
Comandi modificano lo stato
(non lo creano)

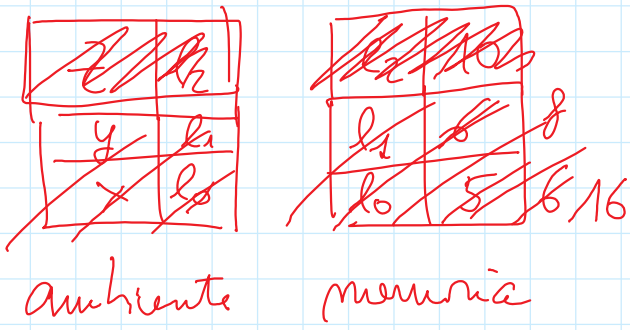
lo stato creato all'interno di un blocco viene CANCELLATO quando il blocco termina (le sue esecuzione)

```
{ int x = 5;
  int y = 6;
  x = x + 1;
  y = y + 2;
}
```



```
{ int x = 5;
  int y = 6;
  x = x + 1;
  { int z = 10;
    x = x + z;
  }
  y = y + ?;
}
```

ANNIDATI



giovedì 20 settembre 2018 16:38

PILA (STACK)

Last In First Out (LIFO)

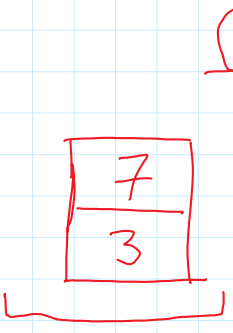
Sequenza di oggetti gestita in modo LIFO

pop (p) → pila (toglie il primo elemento)

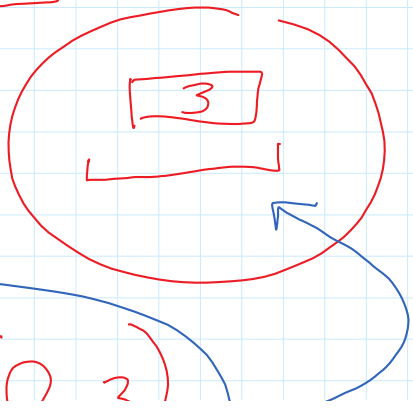
push (p, e) → pila, elemento (aggiunge l'elemento e in testa alla pila p)

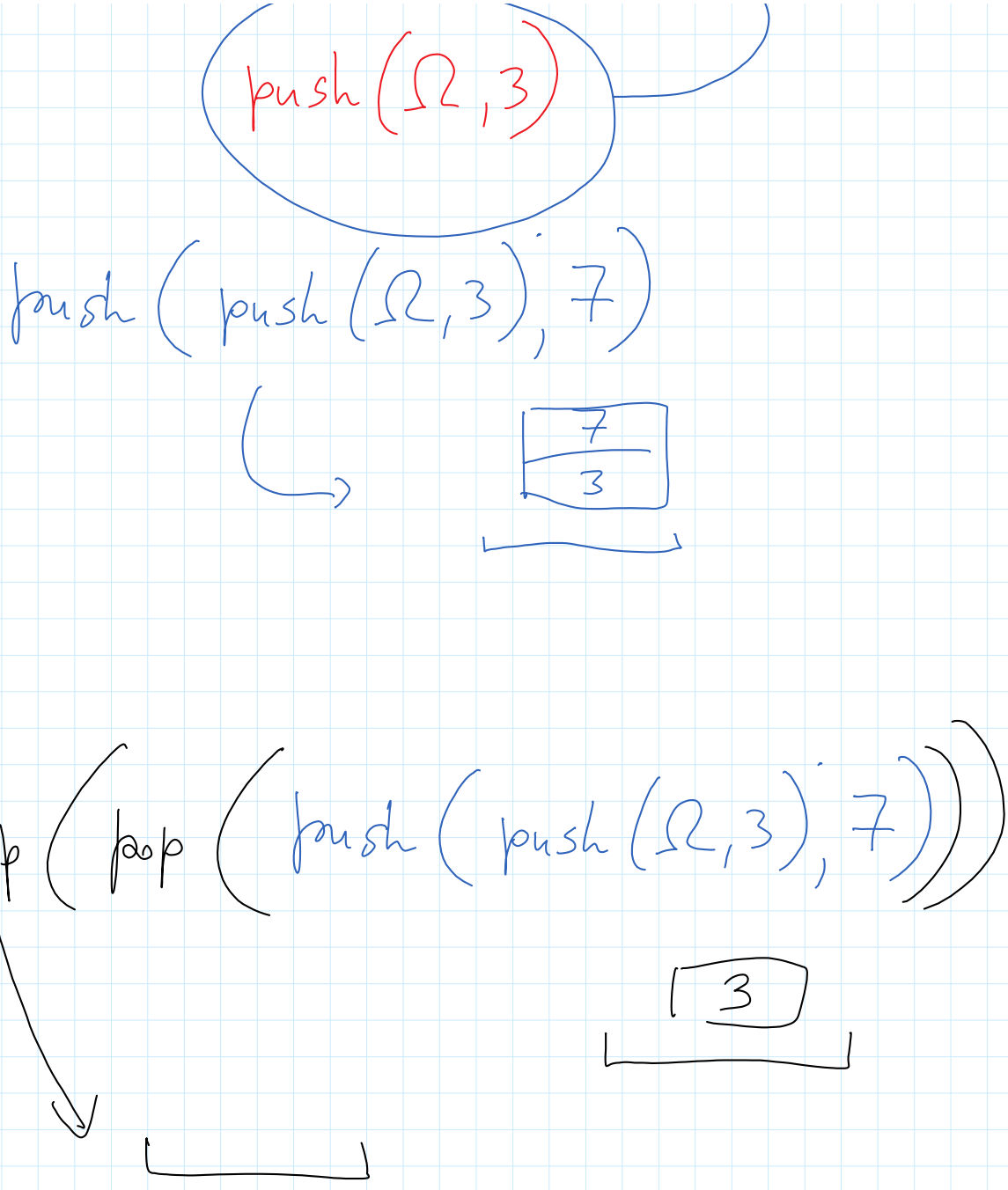
Pila particolare → PILA VUOTA (EMPTY STACK)

Pila di interi



push(Ω, 3)




push(Ω,3)
push(push(Ω,3),7)
7
3
pop(pop(push(push(Ω,3),7)))
3

ambiente

y	l_1
x	l_0

memoria

l_1	6
l_0	5

$\{\langle y, l_1\rangle, \langle x, l_0\rangle\}$ $\{\langle l_0, 5\rangle, \langle l_1, 6\rangle\}$

≠FRAME di ambiente

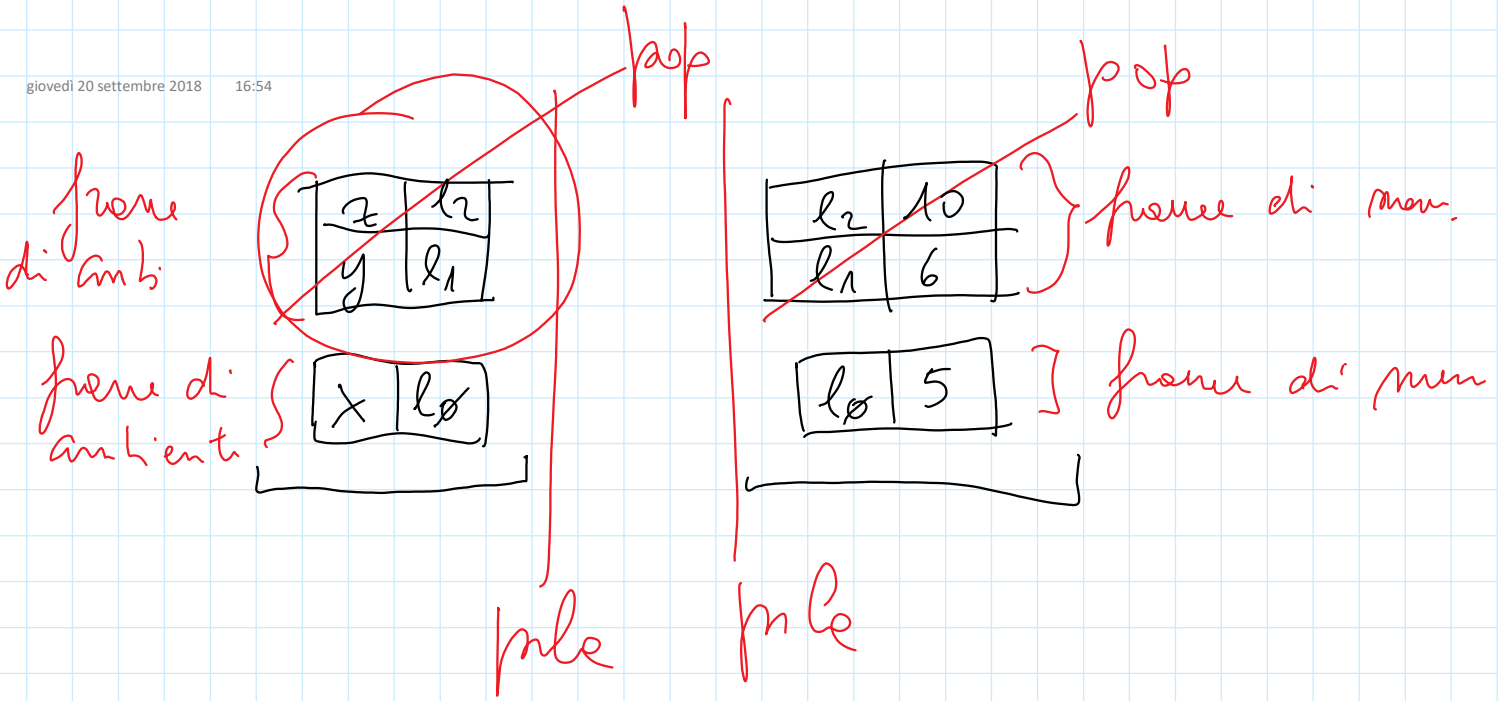
FRAME di memoria

Fino ad adesso lo stato era rappresentato da una coppia

frame di ambiente — frame di memoria

Da ora in poi lo stato sarà rappresentato dalla coppia

PILA di frame di ambiente — PILA di frame di memoria


pop
frame di amb.
z | l2
y | l1
frame di ambiente
x | l0
pile
pop
l2 | 10
l1 | 6
frame di mem.
l0 | 5
frame di mem.
pile

frame particolare - FRAME VUOTO

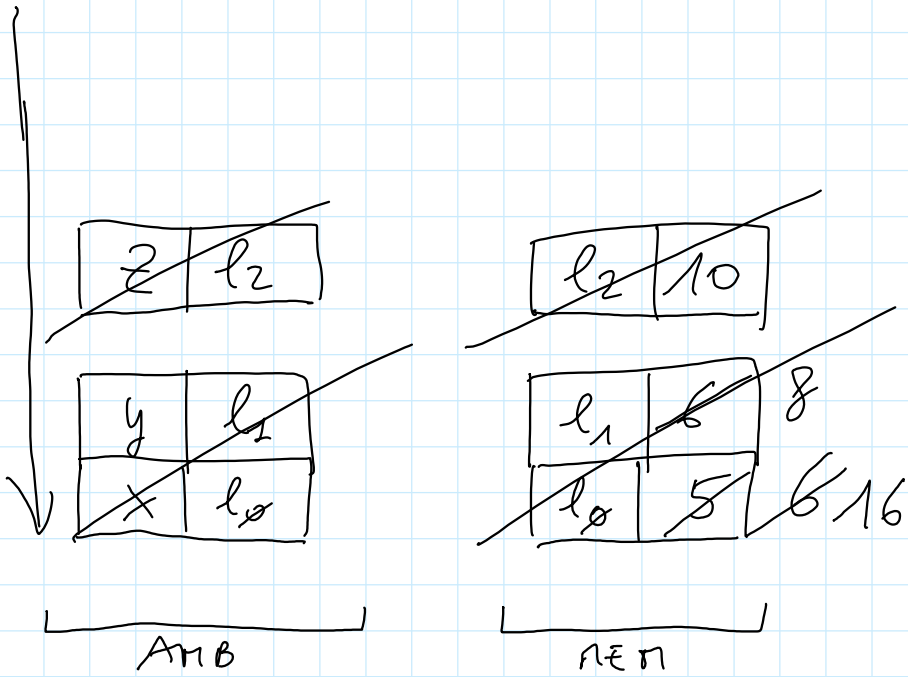
{} ∅

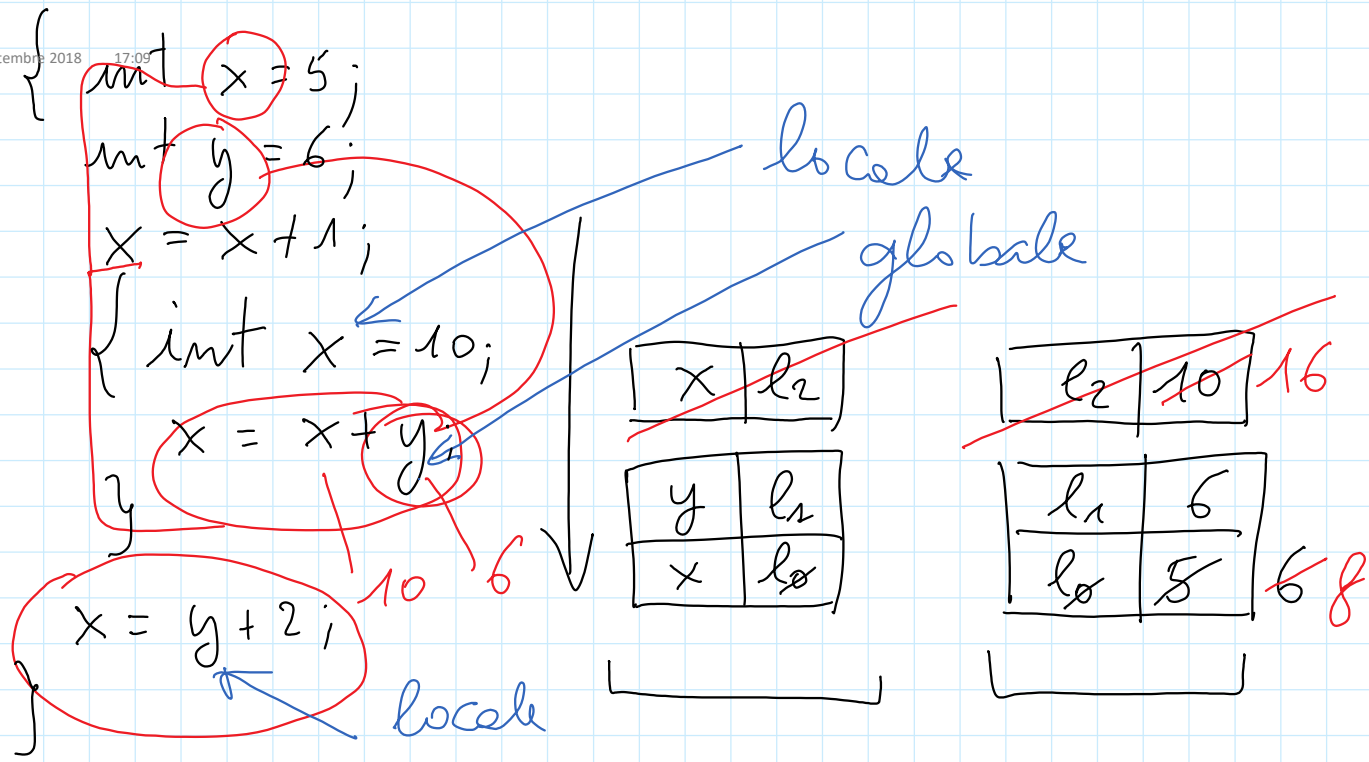
grafo concreto

insieme vuoto

- Quando si inizia l'esecuzione di un blocco si aggiunge (push) un frame vuoto ma nella pila di ambiente che nella pila di memoria
- Le dichiarazioni aggiungono associazioni sul frame in TESTA ma alla pila di ambiente che a quella di memoria
- Quando un blocco termine si fa un'operazione di POP sia sulla pila di amb. che su quella di memoria.

```
{ int x = 5;
  int y = 6;
  x = x + 1;
  { int z = 10;
    x = x + z;
  }
  y = y + z;
}
```





Visibilità degli identificatori (nomi delle variabili)

Un nome (identificatore, variabile) è visibile nel blocco in cui è dichiarato e in tutti i blocchi più interni, a meno che non venga dichiarato di nuovo in un blocco interno.

Dichiarazioni
main ()
Blocco

} programma C

```
int x = 4;
int y = 6;

main()
{
  int i;
  int fact = 1;
  for (i = x; i > 1; i = i - 1)
      fact = fact * i;
  printf( ...., fact);
  fact = 1;
  for (i = y; i > 1; i--)
      fact = fact * i;
  .
  .
  .
```

i--

rendendo parametrico dargli un nome

giovedì 20 settembre 2018 17:26

FUNZIONI e PROCEDURE

- FUNZIONI: modificano lo stato e restituiscono un valore
- PROCEDURE: modificano lo stato

$f(n) = 3 \cdot n$ — nome, parametro

$f : \mathbb{N} \to \mathbb{N}$

$f(5)$ — argomento

uno

$f(n, m) = n^2 \cdot m^2 + 3n + 4m \cdot n$

$f(3, 4)$

giovedì 20 settembre 2018 17:30

tipo del risultato

nome

```
int fact ( int n )
{ int i;
  int f = 1;
  for (i = n; i > 1; i--)
      f = f * i;
} return f;
```

← intestazione

↖ espressione che dà il risultato

```
int fact (....) {...}
int x = 4;
int y = 6;
main ()
{ printf(..., fact(x));
  printf(..., fact(y))
```

← x = fact(x);