

# Sintassi: istruzioni del C++

- **input /output:** << e >> cout e cin (c'è molto di + nel testo 3.1)
- **assegnazione:** assegna un valore ad una variabile
- **istruzione condizionale:** esegue un confronto e a seconda del risultato fa cose diverse
- **iterazione:** ripete alcune istruzioni per tutto il tempo che una condizione è verificata

# assegnazione

$x = \text{ESPRESSIONE}$

$x = a + 12 * \text{bib};$

si calcola il valore  $V$  dell'espressione

servono gli R-valori di  $a$  e  $\text{bib}$   
(eventualmente convertiti)

$V$  viene messo nella RAM nelle posizioni  
indicate dall'L-valore di  $x$  ( $\&x$ )

$x = x+1;$

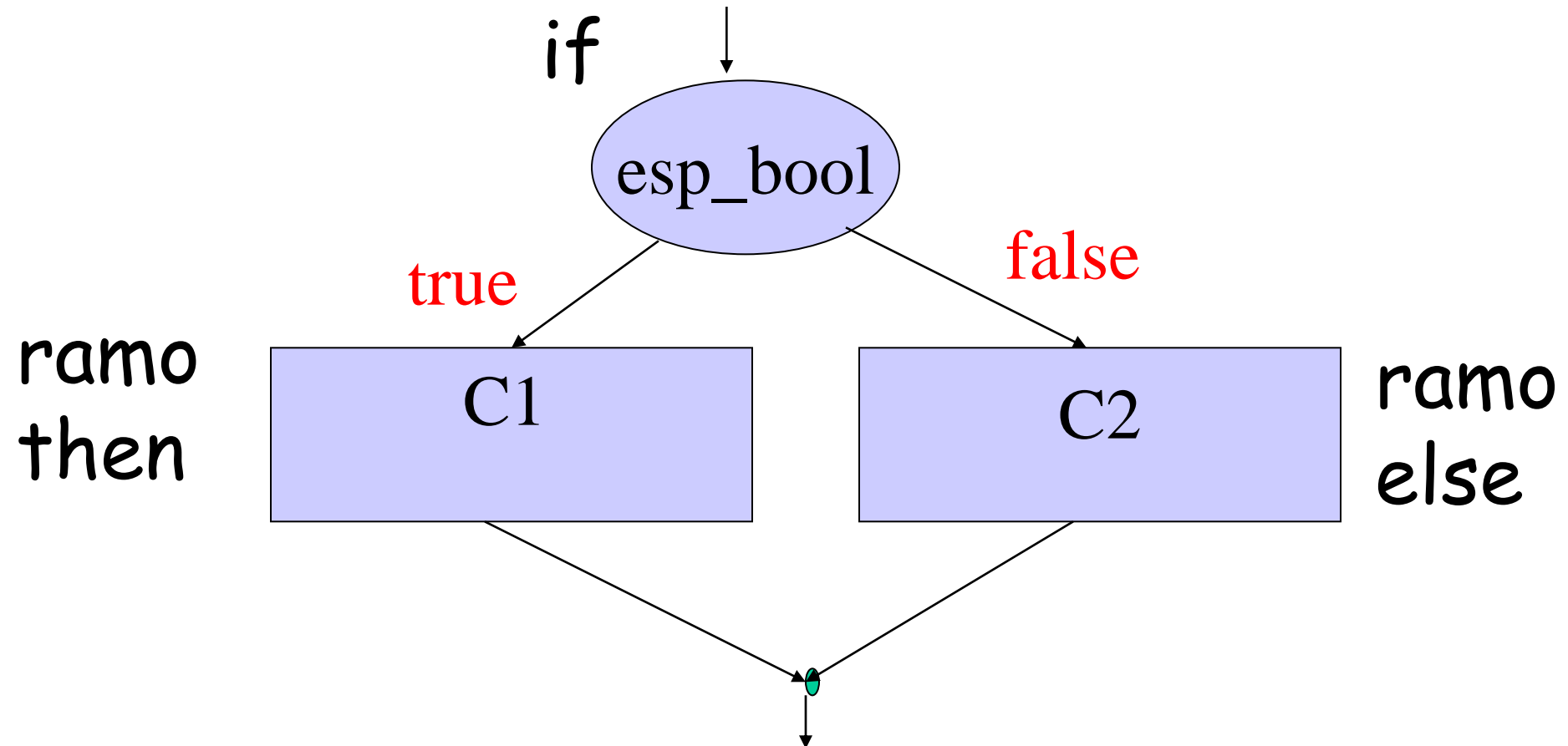
L-valore di  $x$

R-valore di  $x$

perché è a  
quell'indirizzo  
che devo  
mettere il valore  
di  $x+1$

per fare la  
somma

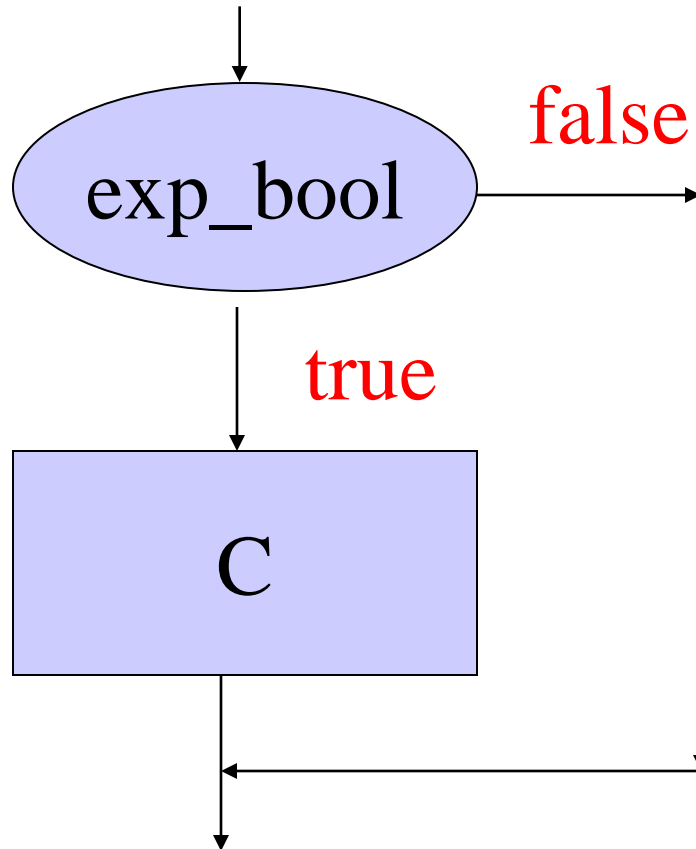
# condizionale o if-then-else



1 punto d'entrata ed 1 d'uscita

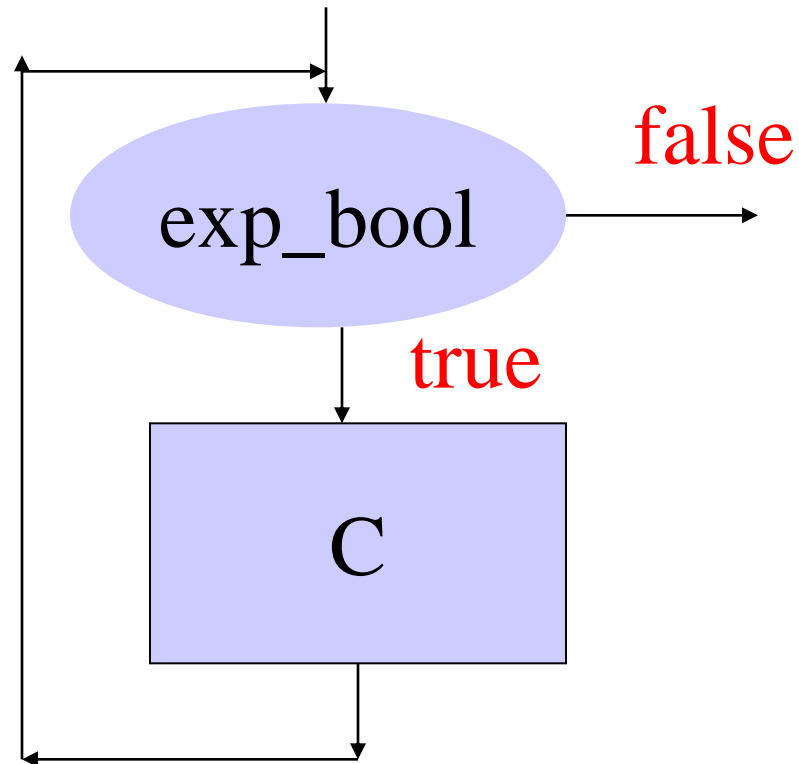
anche: **if(exp\_bool) C;**

solo il  
ramo then



```
main()
{int x, y;
cout<<"inserire 2 interi";
cin >> x >> y;
cout<<" valore di x="<<x<<"valore di y="<<y;
int SOM=x+y;
if(SOM>0)
    y=SOM;
else
    x=SOM;
cout<<" valore di x="<<x<<"valore di y="<<y;
}
```

# Iterazione while



1 punto d'entrata ed 1 d'uscita



esempio di while:

```
int x=0;
```

```
while(x < 10)    {0<= x <=10}
```

```
{                {0<= x <10}
```

```
    x=x+1;
```

```
}    {0<= x <=10} && { x >=10}
```

```
cout << x; // quanto vale x?
```

## esercizio risolto 3.3

leggere da cin 10 interi e stampa il minimo e il massimo tra i valori letti.

idea leggere gli interi ed aggiornare 2 variabili min e max in modo che abbiano i valori giusti per gli interi letti fino a quel momento

un problema generale: inizializzare min e max

```


#include<iostream>
using namespace std;
main()
{int , i=0, max=MIN_INT, min=MAX_INT;
while(i<10)
{cout<<"inserire il prossimo intero ";
cin>> x;
if(max<x)
    max=x;
else
    if(min>x)
        min=x;
i=i+1;
}
cout<<"max="<<max<<" min="<< min<<endl;
}

```

il programma è sbagliato, trovare  
10 interi per cui non calcola i  
valori giusti

correzione:

```
#include<iostream>
using namespace std;
main()
{int , i=0, max=MIN_INT, min=MAX_INT;
while(i<10)
{cout<<"inserire il prossimo intero ";
cin>> x;
if(max<x)
    max=x;
else
    if(min>x)
        min=x;
i=i+1;
}
cout<<"max="<<max<<" min="<<min<<endl;
}
```



```
if(max < x)
{
    max=x;
    if (i==0)
        min=x;
}
```

ma siamo sicuri  
che sia corretto? E  
il ramo else ?

altra soluzione a pag 29 del testo (SOLUZIONE 2)

## Esercizio 4.2:

dato  $X \geq 1$  vogliamo calcolare il minimo  
esponente tale che  $2^{\text{esponente}} \geq X$

PRE=(cin contiene intero >0)

```
int X, potenza=1, esponente=0;  
cin >> X;  
while( X > potenza) R = invariante  
{  
    potenza=potenza*2;  
    esponente = esponente + 1;  
}  
cout<< "l'esponente e'=" << esponente<<endl;
```

POST= (esponente è il minimo esponente  
tale che  $2^{\text{esponente}} \geq X$ )

# L'invariante R

R=

(  $X > 0$ ,  
potenza=2 esponente ,  
 $X > 2^{(\text{esponente}-1)}$   
)



esercizio risolto 4.3: si legge sequenza di caratteri numerici fino a leggere 'a'

esempio: '3' '2' '6' 'a'

il programma deve calcolare l'intero 326

PRE=(cin contiene  $c_1 \dots c_k$  'a',  $n \geq 0$ ,  $c_j$  è un carattere numerico, per ogni  $j \in [1, k]$ )

programma ?

POST=(calcola NUM( $c_1 \dots c_k$ ))

NUM('2' '3' '1')=231    NUM()=0

inizio :

leggi un carattere

se è numerico fai i conti e torna all'inizio

altrimenti fine

fai i conti ? 

se abbiamo letto

$c_1c_2c_3$  abbiamo  $NUM(c_1c_2c_3)$  e se  $c_4 \neq 'a'$   
dobbiamo calcolare  $NUM(c_1c_2c_3c_4)$

come ??

$$NUM(c_1c_2c_3c_4) = 10 * NUM(c_1c_2c_3) + NUM(c_4)$$

```
char q; int num=0, n=1;  
IN>>q;
```

```
R=(num=NUM(c1..c(n-1)), q=cn)
```

```
while(q != 'a')  
{  
    num=num*10+(q-'0');  
    IN >> q;  
    n++;  
}
```

Esercizio risolto 4.7: leggere valori interi da IN e scriverli su OUT

al più 10 valori, ma

fermarsi con due 0 consecutivi **senza  
scrivere**li

PRE=( IN=b1...bk, k>9 o contiene due 0 consecutivi)

programma ?

POST=(OUT=Prefix(b1...bk))

Prefix(b1..bk) = b1...b10 se b1..b10 non  
contiene due 0 consecutivi, altrimenti  
Prefix(b1...bj00...)=b1...bj

se leggo 0 non lo posso stampare

situazione a inizio ciclo:

- se ho letto 2 zeri consecutivi devo terminare,

- altrimenti **devo distinguere:** se ho letto uno zero al giro precedente o no

idea: due booleani per distinguere

- ho letto uno zero o no

- ho letto due zeri o no

## invariante R

1.  $0 \leq n \leq 10$
2. letti  $b1 \dots bn$
3.  $(!uno0 \ \&\& \ !due0) \Rightarrow (OUT = b1 \dots bn)$
4.  $due0 \Rightarrow (bn-1 = bn = 0 \ \&\& \ OUT = b1 \dots bn-2)$
5.  $(uno0 \ \&\& \ !due0) \Rightarrow (bn = 0 \ \&\& \ OUT = b1 \dots bn-1)$



```

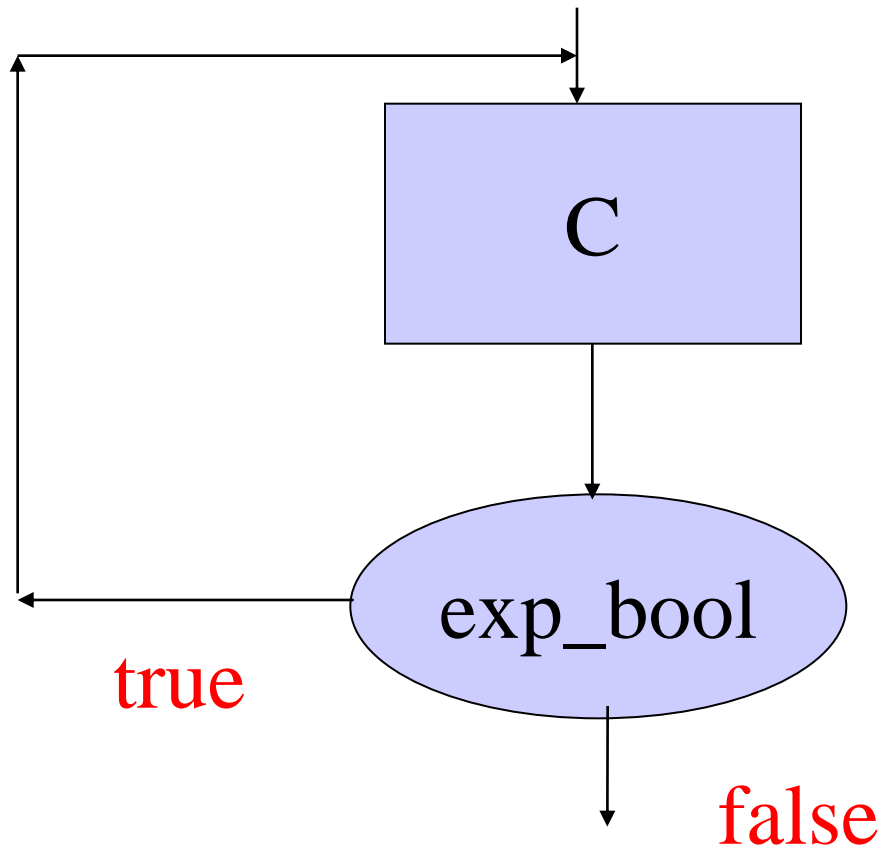
int X, n=0;
bool uno0=false, due0=false;
while(! due0 && n < 10)
{ IN >> X; n++;
  if(X==0)
    if(uno0)
      due0=true;
    else
      uno0=true;
  else
    {if(uno0)
      {OUT<<0 <<' '; uno0=false;}
      OUT << X <<' ';
    }
  if(uno0 && !due0) OUT << 0;
}

```

manca:

```
if(uno0 && !due0)  
    OUT<< 0;
```

do-while: prima  $C$  e poi il test



1 punto d'entrata ed 1 d'uscita

esempio di do\_while:

```
int x=0;
```

```
do    {0<= x < 10}
```

```
x=x+1; {0<= x <= 10}
```

```
while(x<10); {0<= x <= 10} && {x >=10}
```

```
cout << x;   {x = 10}
```