

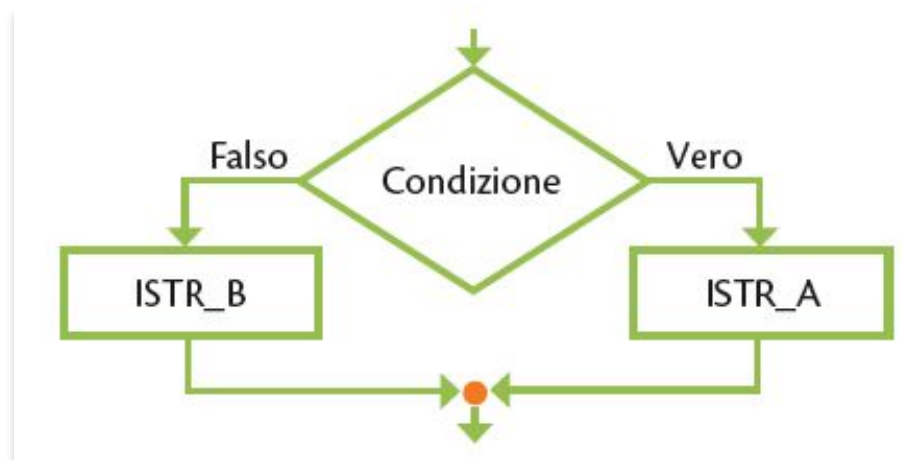
WEB DEVELOPER BASI DEL CODING Massimo PAPA La Selezione

La selezione

La struttura di selezione

La **struttura di selezione** permette di effettuare una scelta fra due possibili alternative, valutando una condizione.

Se la **condizione** è vera vengono eseguite le **istruzioni** presenti sul ramo corrispondente al vero, altrimenti vengono eseguite le condizioni presenti sul ramo del falso.



La selezione in C++ (1)

Con riferimento a C++ l'istruzione condizionale può avere diverse forme.

```
if (condizione)
    istruzione;
```

```
if (condizione)
    istruzione_1;
else
    istruzione_2;
```

Un altro utile costrutto C++ per realizzare strutture condizionali con più possibilità di scelta è quello della **condizione multipla**, realizzata usando **switch case**.

La selezione in C++ (2)

Nella tabella elenchiamo le varie modalità per scrivere le istruzioni di selezione.

Simbolo	Significato	Simbolo	Significato
=	assegnazione	&	AND binario
==	uguale	>	maggiore
!=	diverso	<	minore
	OR logico	>=	maggiore o uguale
&&	AND logico	<=	minore o uguale
	OR binario		

Le selezioni semplici: un esempio – problema e analisi

Il problema

Dato in un numero calcola il cubo se è maggiore di zero, il quadrato in caso contrario.

L'analisi

Occorre per prima cosa prendere in input un numero e poi effettuare un confronto del valore di tale numero con zero.

Se il numero è maggiore di zero, lo si moltiplica 3 volte per sé stesso, altrimenti lo si moltiplica solo 2 volte.

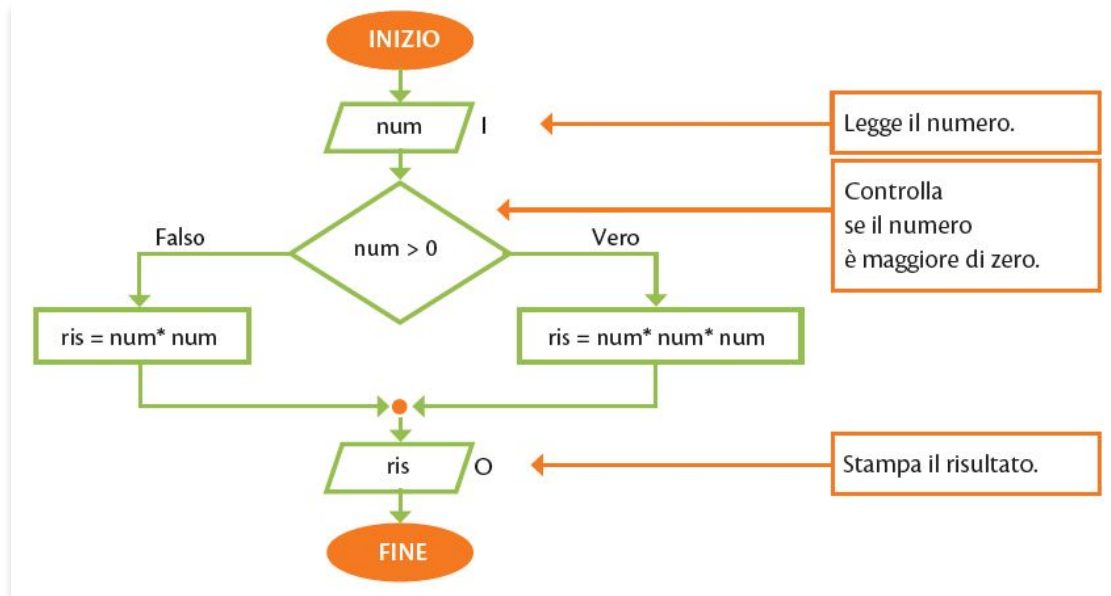
Il risultato sarà sicuramente un numero positivo (o uguale a zero, se il numero di partenza è zero).

Le selezioni semplici: un esempio – variabili e algoritmo

Le variabili

Nome	Tipo	Utilizzo	Descrizione
num	reale	input	numero in input
ris	reale	output	risultato

L'algoritmo



Le selezioni semplici: un esempio – test e simulazione

Il test e la simulazione

Verifichiamo la correttezza del programma effettuando la simulazione con due diversi casi prova.

Si deve infatti testare sia il caso in cui il numero in input è positivo, sia quello in cui il numero è negativo.

Caso prova 1

$\text{num} \leftarrow 3$

$3 > 0$? Vero

$\text{ris} \leftarrow 3 * 3 * 3 = 27$

27 (ris) in output

Caso prova 2

$\text{num} \leftarrow -2$

$-2 > 0$? Falso

$\text{ris} \leftarrow -2 * -2 = 4$

4 (ris) in output

Le selezioni semplici: un esempio – codifica in C++ ed esecuzione

La codifica in C++

```
#include <iostream>
using namespace std;
int main (){
    float num;
    float ris;
    cout<<"inserire un numero"<<endl;
    cin>>num;
    if (num > 0)
        ris =num*num*num;
    else
        ris = num*num;
    cout << "risultato = " << ris <<endl;
    system ("pause");
    return(0);
}
```

L'esecuzione

Caso 1

```
inserire un numero
3
risultato = 27
Premere un tasto per continuare . . . _
```

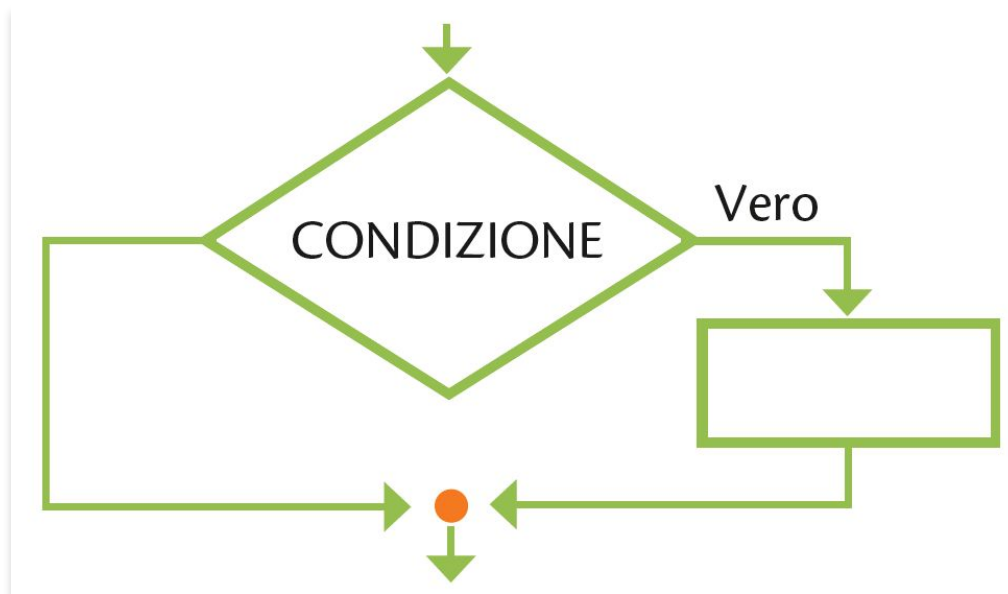
Caso 2

```
inserire un numero
-2
risultato = 4
Premere un tasto per continuare . . . _
```


Le selezioni a una via

Quando sono presenti istruzioni solo in un ramo della condizione si parla di **condizione a una via**.

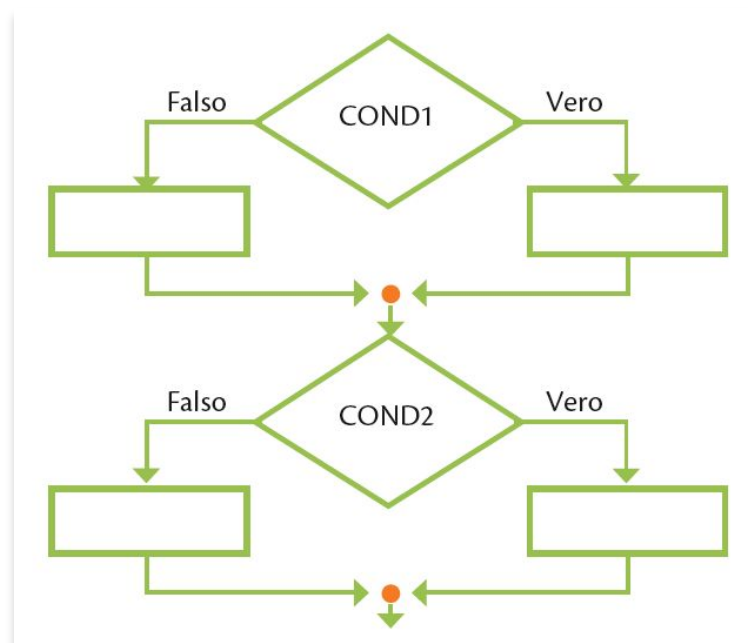
È buona norma inserire le istruzioni da eseguire sul ramo del vero, lasciando vuoto il ramo del falso.



Le selezioni in cascata

Quando due o più condizioni devono essere eseguite una dopo l'altra si parla di **selezioni in cascata**.

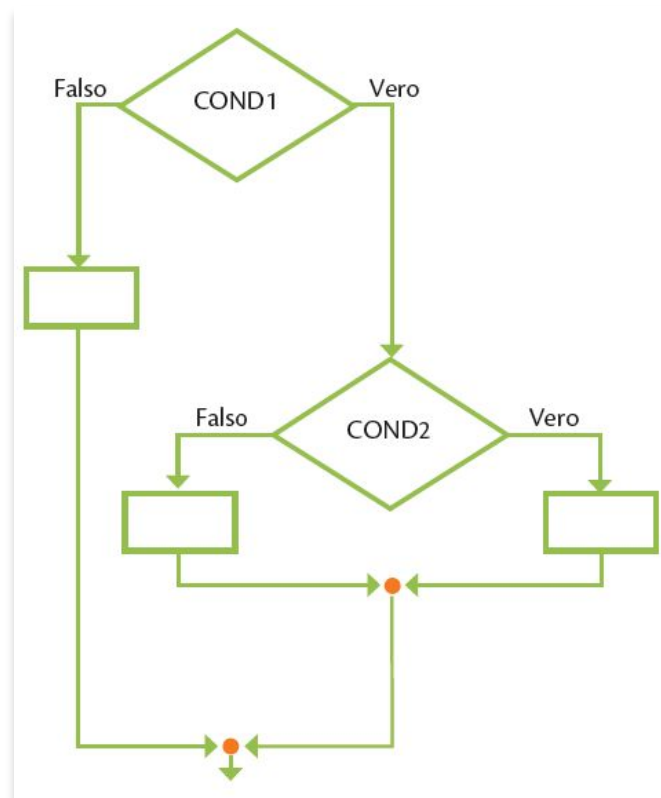
Prima viene testata la prima condizione e di seguito la seconda, indifferente dal risultato della prima.



Le selezioni annidate

Quando nel ramo di una condizione è presente un'altra condizione si parla di **selezioni annidate**.

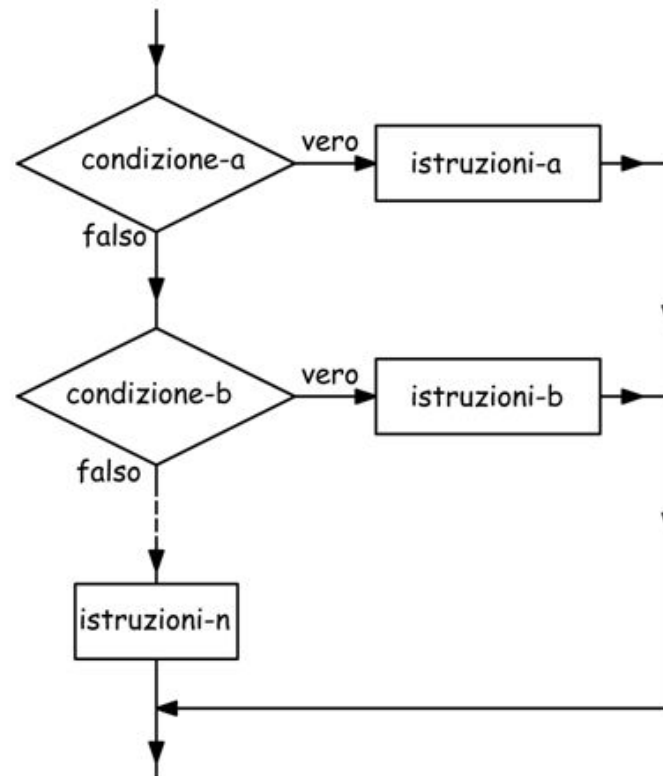
La condizione presente sul ramo del vero (o del falso) di una precedente condizione viene presa in considerazione solo se la condizione esterna risulta vera (o falsa).



Le selezioni annidate

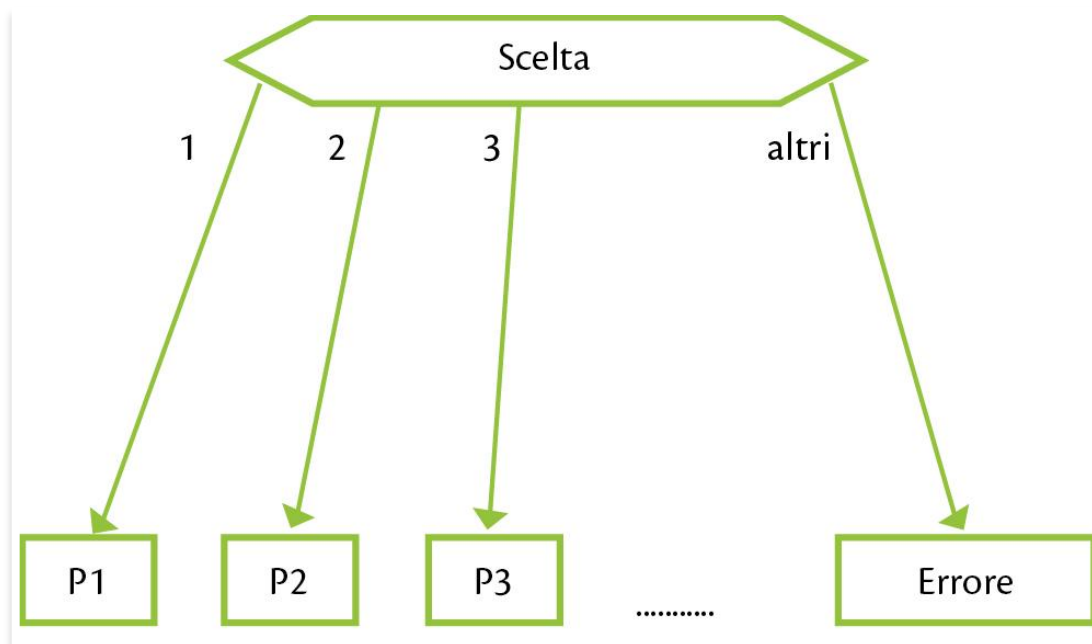
Quando nel ramo di una condizione è presente un'altra condizione si parla di **selezioni annidate**.

```
if(condizione-a){istruzioni-a;}  
else if(condizione-b){istruzioni-b;}  
else if(condizione-c){istruzioni-c;}  
...  
else {istruzioni-n;}
```



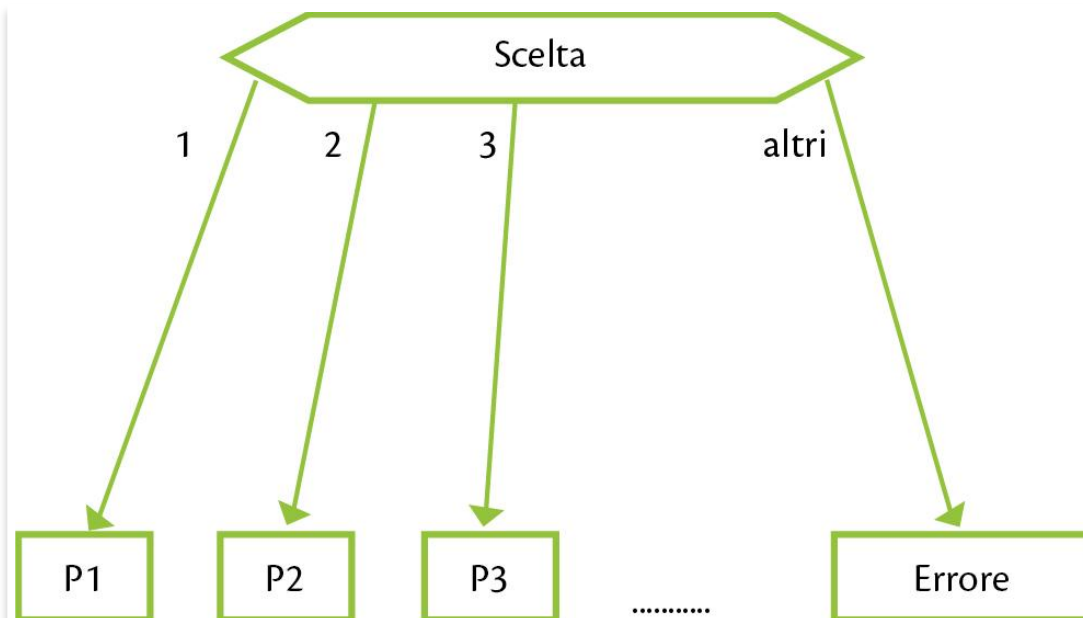
La selezione multipla

Il costrutto di **selezione multipla** è un blocco di condizione che presenta più di due uscite, semplificando così la scrittura di istruzioni condizionali annidate.



La selezione multipla

Si codifica con il costrutto **switch - case**



```
int scelta;  
switch(scelta){  
    case 1:  
        //Elab. 1  
        ----  
        break;  
    case 2:  
        // Elab. 2  
        ----  
        break;  
    default:  
        // Elab. errore  
        ----  
        break;  
}
```

Concetti di logica

Una **proposizione** è un costrutto linguistico che può assumere il valore vero oppure il valore falso.

La verità o la falsità della proposizione è detta **valore di verità**.

Una proposizione può essere vera o falsa, ma non entrambe le cose. I due valori **vero** o **falso** sono detti **valori logici** o **booleani**.

Le proposizioni possono anche essere combinate tra loro tramite alcuni operatori (**connettivi**), in modo da formare una nuova proposizione, che è detta **composta**.

Concetti di logica: il connettivo and

Due proposizioni possono essere collegate dalla particella «e» (**and**), così da formare una proposizione composta, detta **coniunzione** delle proposizioni di partenza.

Se indichiamo con **p** e **q** le proposizioni, abbiamo:

p and q

Il valore di verità della proposizione composta *p and q* è dato dalla **tavola di verità**, mostrata nella tabella qui a fianco.

p	q	p and q
v	v	v
v	f	f
f	v	f
f	f	f

Concetti di logica: il connettivo or

Due proposizioni possono essere collegate dalla particella «o» (**or**), così da formare una proposizione composta detta **disgiunzione** delle proposizioni di partenza.

Se indichiamo con **p** e **q** le proposizioni, abbiamo:

p or q

Il valore di verità della proposizione composta **p or q** è dato dalla **tavola di verità**, mostrata nella tabella qui a fianco.

p	q	p or q
v	v	v
v	f	v
f	v	v
f	f	f

Concetti di logica: il connettivo not

Data una proposizione p è possibile formare un'altra proposizione, detta la **negazione** di p , scrivendo «è falso che...» prima di p oppure, quando è possibile, inserendo in p la parola «non» (**not**).

I simboli « p' », « $\neg p$ » o ancora « \hat{p} » denotano la negazione di p .

Il valore di verità della proposizione composta p' è dato dalla **tavola di verità**, mostrata nella tabella qui a fianco.

p	$\text{not } p$
v	f
f	v

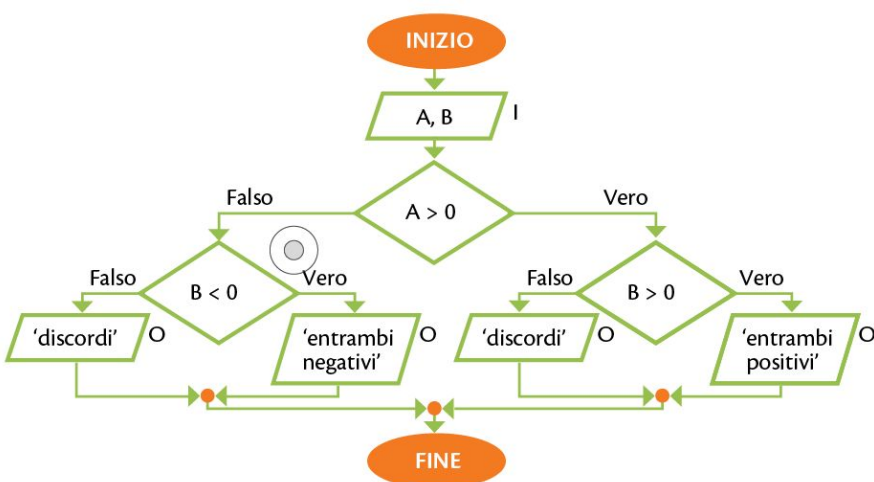
L'uso dei connettivi logici

Utilizzando i **connettivi logici** *and*, *or* e *not*, è possibile ridurre la complessità degli algoritmi inserendo in essi condizioni complesse.

Esempio

Dati due numeri, dire se sono entrambi positivi o negativi.

Soluzione senza connettivi logici



Soluzione con connettivo *and*

