



Programmazione strutturata


Francesco Isgrò


Contiene materiale dalle lezioni di
Alessandro Ursomando



C

- Linguaggio di programmazione
- Creato da Dennis Ritchie (Bell labs) nel 1972
- Usato come linguaggio per Unix
- Prima versione di Unix da Ken Thompson (creatore di Unix) basata su
 - Assembly
 - Linguaggio B
- C creato per superare alcune limitazioni di B


- 
- Linguaggio B
 - Creato nel 1967
 - Non tipizzato
 - Usava la machine word come tipo di dato
 - Uso pesante di puntatori e indirizzi
 - C: evoluzione B incorporando la tipizzazione
 - Rispetta i canoni della programmazione strutturata

- 
- Traditional C (inizio anni 80)
 - Tipo *void*
 - Tipo *enum*
 - ANSI C (fine anni 80)
 - Tipo *void **
 - Prototipi delle funzioni
 - Definizione del linguaggio più precisa



ANSI


- American National Standards Institute
- Definisce standard per molti tipi di sistemi, inclusi linguaggi di programmazione
- Standard definiti per il C approvati dall'ISO (International Organization for Standardization) nel 1990
- ANSI C riconosciuto standard internazionale

- 
- Lo standard specifica
 - Il formalismo del linguaggio
 - La semantica del linguaggio
 - Lo scopo dello standard
 - Portabilità
 - Affidabilità
 - Manutenibilità



Programmazione strutturata

- Dijkstra (1969)
- Proposta per regolamentare e standardizzare le metodologie di programmazione.
- Obiettivo: rendere più facile
 - lettura dei programmi
 - modifica dei programmi
 - manutenzione dei programmi

- 
- Critica alla sequenza e all'uso del GOTO
 - Bassa qualità e leggibilità del codice (*spaghetti code*)
 - Teorema di Bohm-Jacopini: qualsiasi programma scritto usando il goto può essere riscritto senza, a patto di avere a disposizione altri tre tipi di strutture di controllo: sequenza, selezione e iterazione.

PASSI ELEMENTARI E STRUTTURE DI CONTROLLO DEL FLUSSO

Le strutture di controllo del flusso sono tre.



I passi vengono eseguiti uno dopo l'altro incondizionatamente

SEQUENZA



I passi vengono eseguiti solo dopo avere verificato una condizione

SELEZIONE



I passi elementari vengono eseguiti più volte

ITERAZIONE

Vediamole nel dettaglio nelle sezioni che seguono.



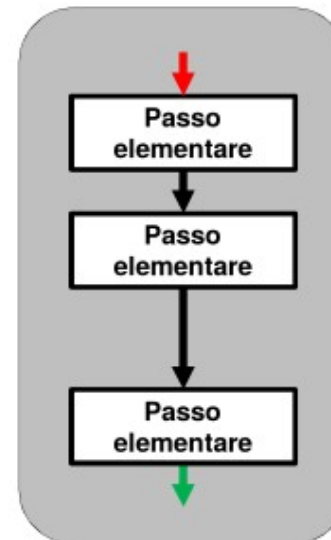
SEQUENZA

La **sequenza** è la struttura di controllo più banale, vediamo qualche flow chart di esempio.



I passi vengono
eseguiti uno dopo
l'altro
incondizionatamente

SEQUENZA



SEQUENZA

Chiedere informazioni.



PROBLEMA DEL MONDO REALE

Prosegui diritto per 100 metri

Gira a destra

Prosegui diritto fino alla fine della strada

Gira a destra

Prosegui fino al secondo semaforo

Gira a sinistra

Prosegui diritto fino alla rotatoria

Prendi la terza uscita

Prosegui diritto per 500 metri

SEQUENZA



SEQUENZA

Calcolare l'area di un quadrato.



PROBLEMA DI INFORMATICA

Prendi in input il valore
della variabile **lato**

Calcola il prodotto di
lato per **lato** e metti il
valore nella variabile
area

Metti in output il valore
della variabile **area**

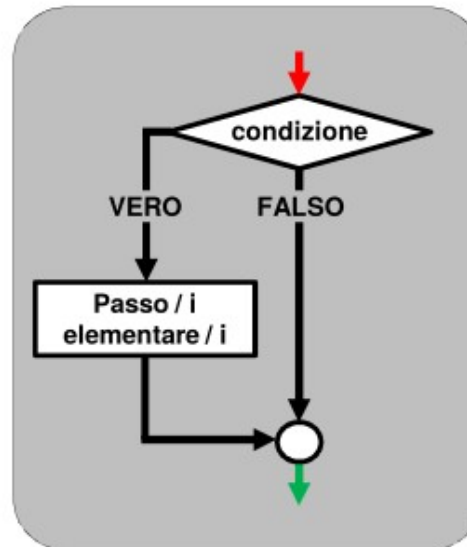
SELEZIONE

La selezione può essere di 3 tipi.
Cominciamo dal più banale: **selezione a una via**.

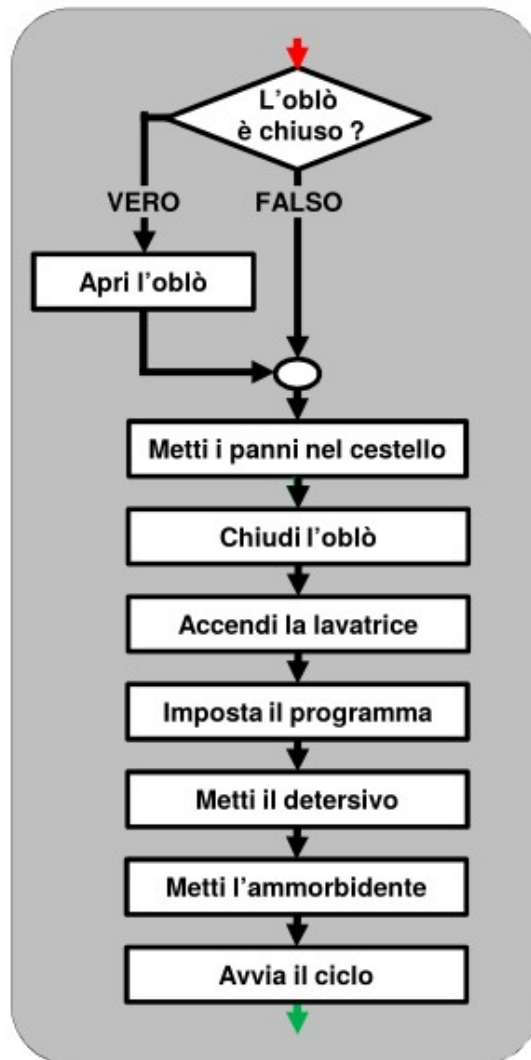


I passi vengono
eseguiti solo dopo
avere verificato una
condizione

SELEZIONE



SELEZIONE A UNA VIA

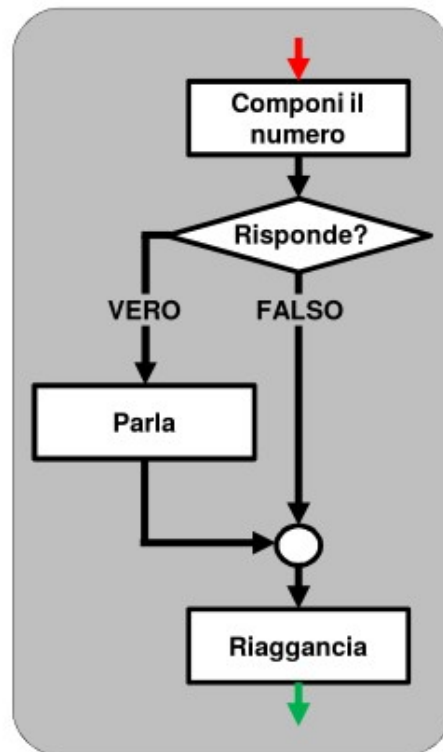


Lavare il bucato con la lavatrice.

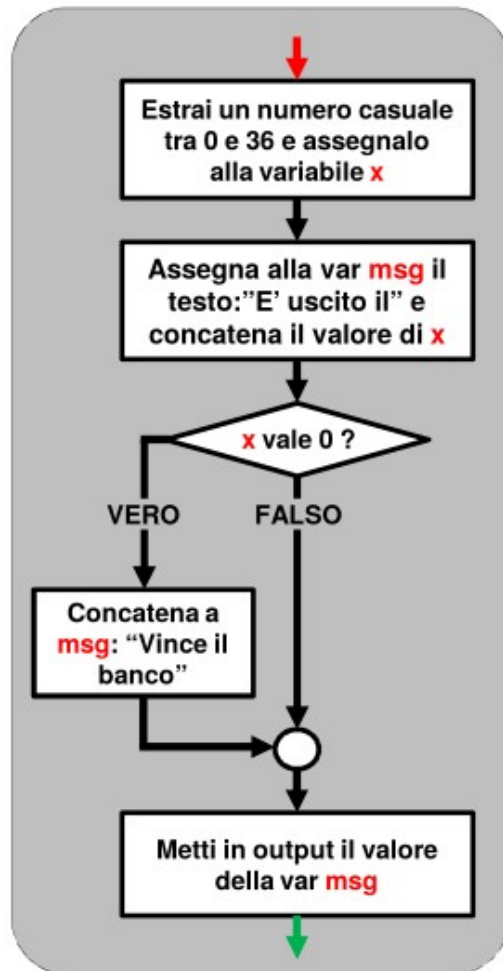


PROBLEMA DEL MONDO REALE

SELEZIONE A UNA VIA



SELEZIONE A UNA VIA



Simulare una sessione di gioco alla roulette.



PROBLEMA DI INFORMATICA

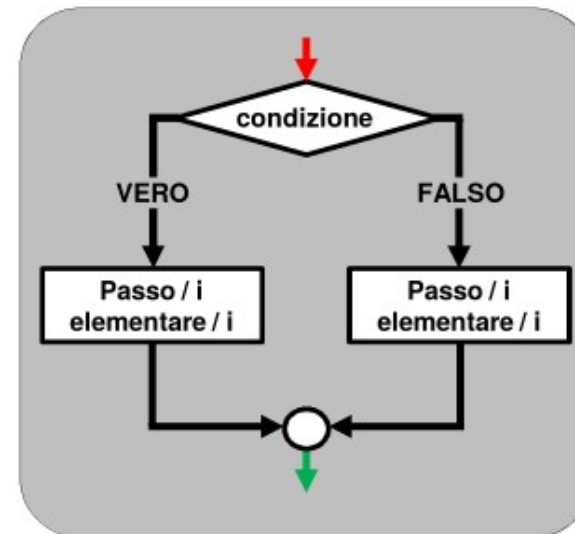
SELEZIONE

La **selezione a due vie** prevede delle operazioni da svolgere sia nel caso in cui la condizione risulti vera, sia nel caso risulti falsa.



I passi vengono eseguiti solo dopo avere verificato una condizione

SELEZIONE

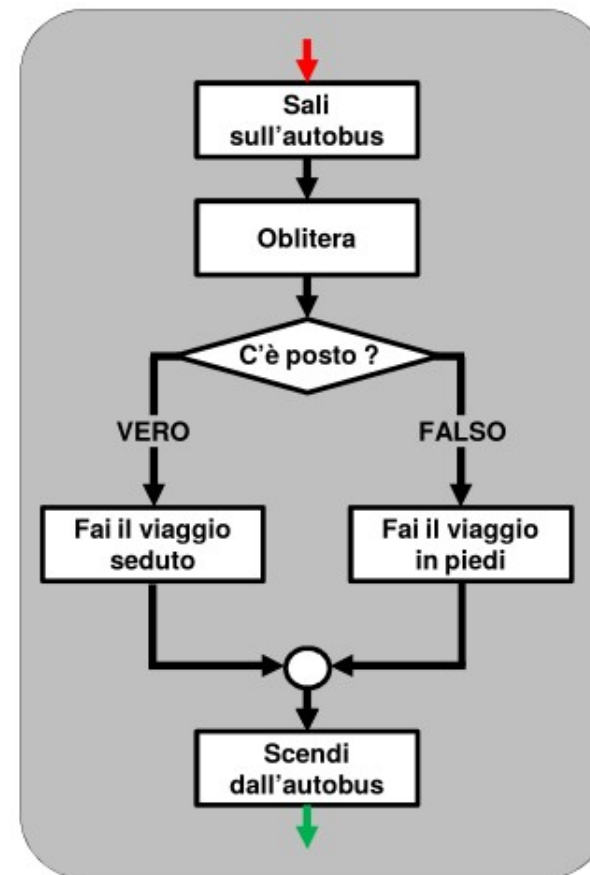


SELEZIONE A DUE VIE

Fare un viaggio in autobus.



PROBLEMA DEL MONDO REALE

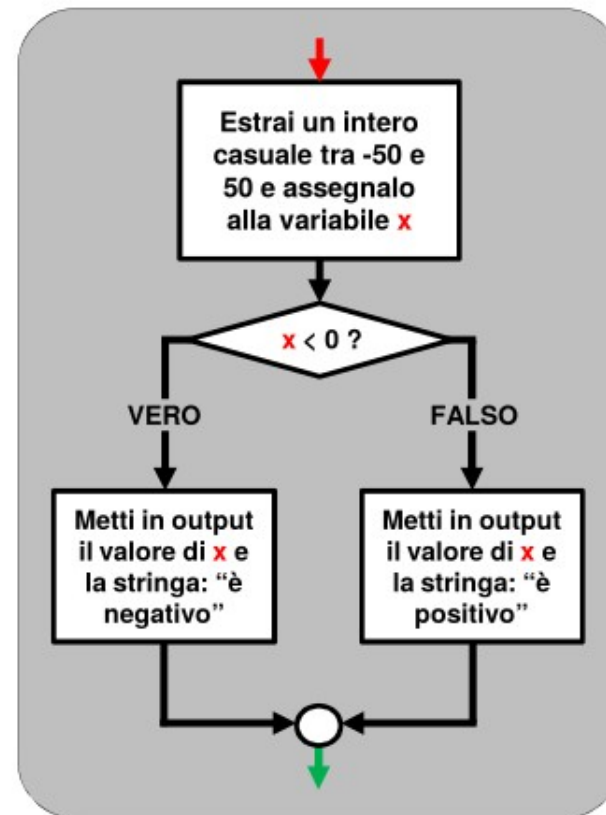


SELEZIONE A DUE VIE

Riconoscere il segno di un numero casuale.



PROBLEMA DI INFORMATICA



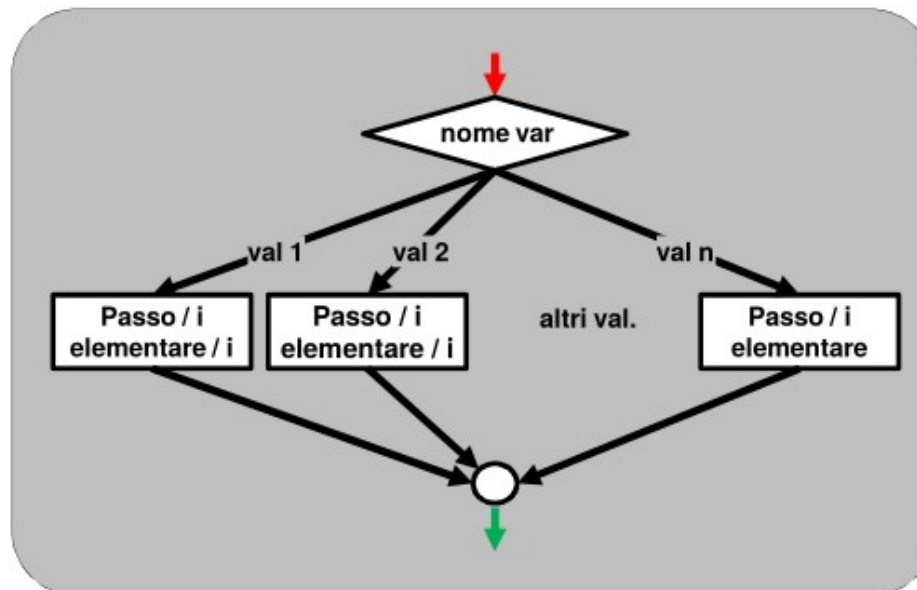
SELEZIONE

La **selezione ad n vie** prevede delle operazioni da svolgere in diversi casi.

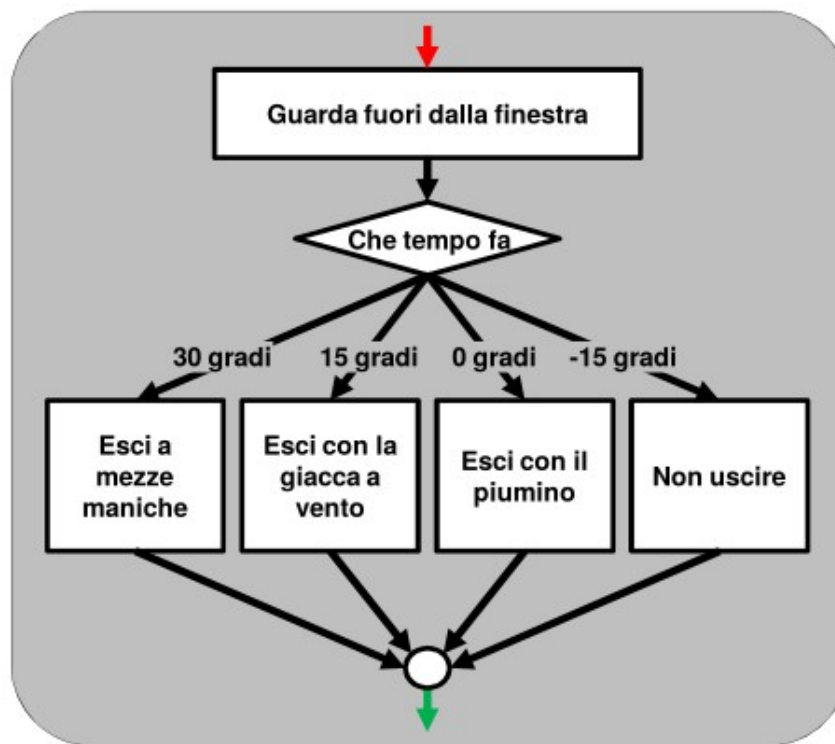


I passi vengono eseguiti solo dopo avere verificato una condizione

SELEZIONE



SELEZIONE AD N VIE

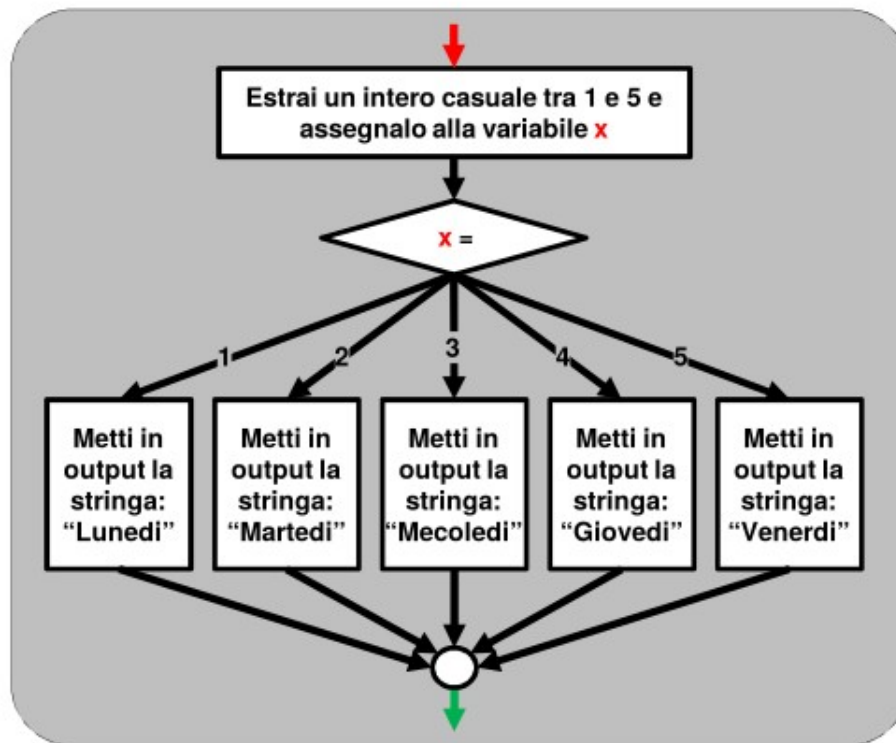


Decidere cosa indossare.



PROBLEMA DEL MONDO REALE

SELEZIONE AD N VIE



Estrarre un giorno lavorativo a caso.



PROBLEMA DI INFORMATICA

ITERAZIONE

Sebbene il concetto di iterazione sia semplice
qui la faccenda si complica perché esistono 3 tipi di cicli.



I passi elementari
vengono eseguiti
più volte

ITERAZIONE

La difficoltà sta
nel capire in che
circostanze è più opportuno
usare l'uno o l'altro.

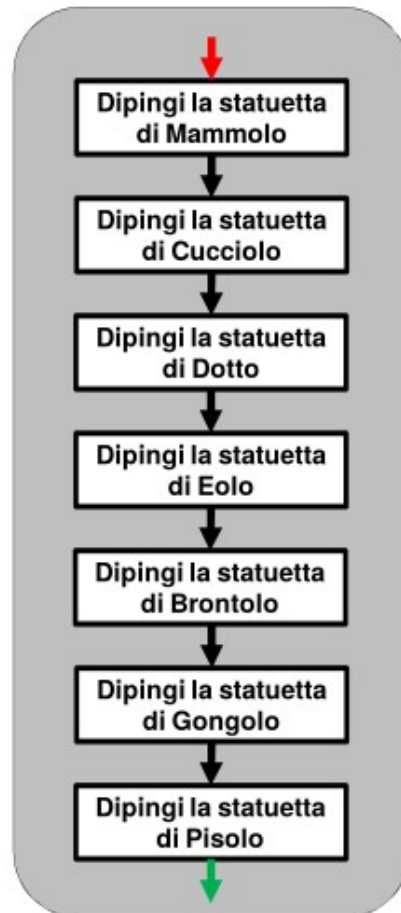


Cominciamo,
come al solito,
dal più semplice.



ITERAZIONE

Il concetto di ciclo è stato introdotto per alleggerire situazioni come questa.

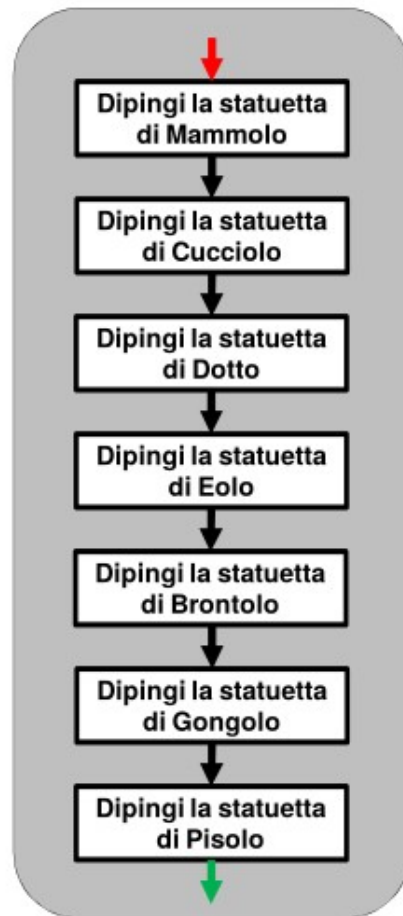


Dipingere le statuette dei nani



PROBLEMA DEL MONDO REALE

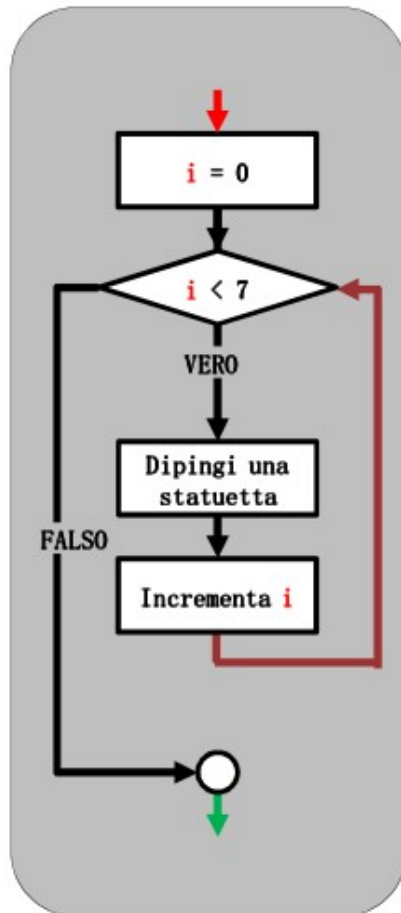
ITERAZIONE



Il concetto di ciclo è stato introdotto per alleggerire situazioni come questa.

Una volta isolata l'istruzione (o le istruzioni) da eseguire **più volte** potremmo scrivere un algoritmo che semplicemente esegua quelle istruzioni (o quella istruzione) quante volte sono necessarie.

ITERAZIONE



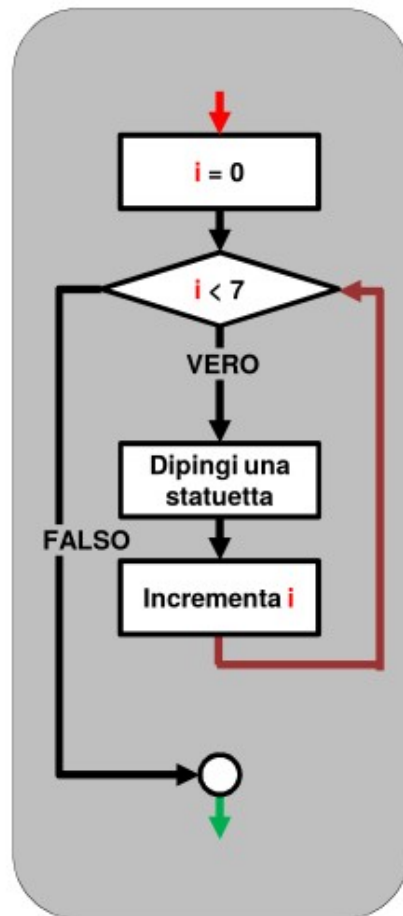
Il concetto di ciclo è stato introdotto per alleggerire situazioni come questa.

Una volta isolata l'istruzione (o le istruzioni) da eseguire **più volte** potremmo scrivere un algoritmo che semplicemente esegua quelle istruzioni (o quella istruzione) quante volte sono necessarie.

Per far ciò useremo una **variabile** con la quale contiamo quante volte abbiamo eseguito il ciclo.

Questo tipo di variabile viene denominata **contatore** e di solito il suo nome è **i**.

CICLO DI RIPETIZIONE SU CONTATORE



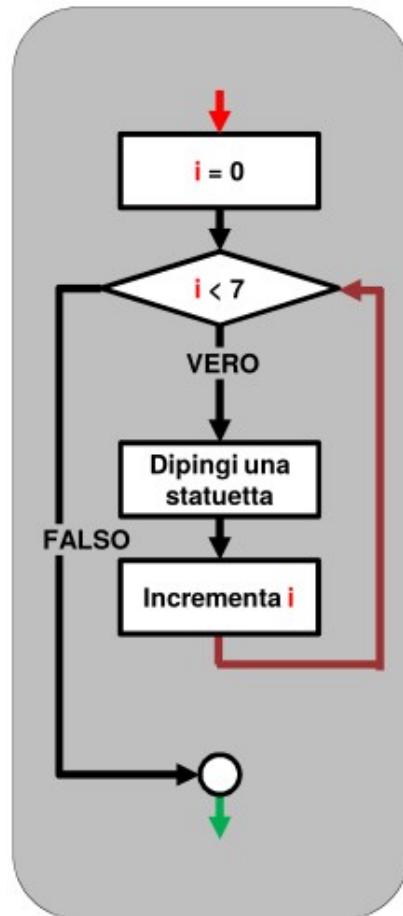
Il concetto di ciclo è stato introdotto per alleggerire situazioni come questa.

Una volta isolata l'istruzione (o le istruzioni) da eseguire **più volte** potremmo scrivere un algoritmo che semplicemente esegua quelle istruzioni (o quella istruzione) quante volte sono necessarie.

Per far ciò useremo una **variabile** con la quale contiamo quante volte abbiamo eseguito il ciclo.

Questo tipo di variabile viene denominata **contatore** e di solito il suo nome è **i**.

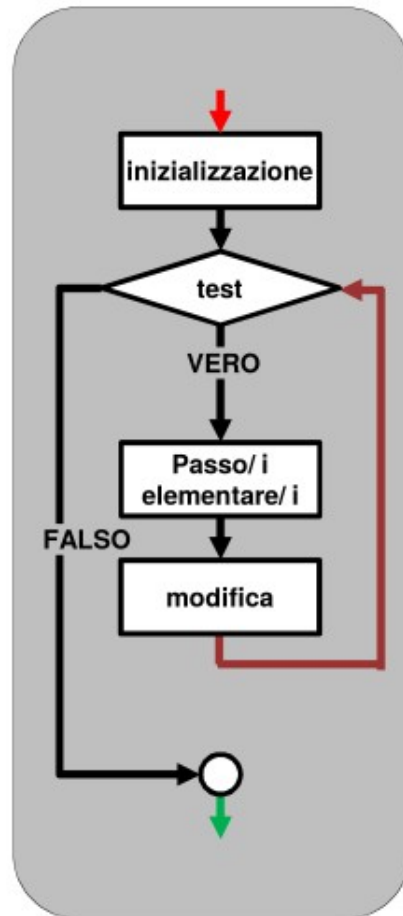
CICLO DI RIPETIZIONE SU CONTATORE



La variabile contatore viene quindi
inizializzata prima di fare il ciclo,
testata prima di ripetere il ciclo e
modificata una volta eseguito il ciclo.



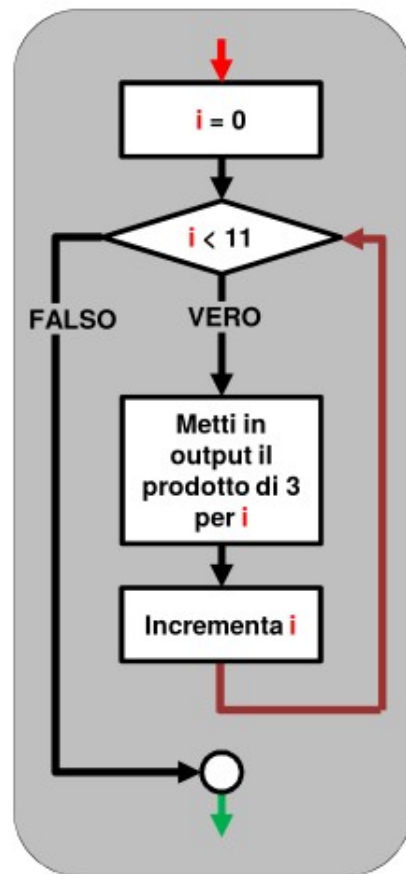
CICLO DI RIPETIZIONE SU CONTATORE



La variabile contatore viene quindi
inizializzata prima di fare il ciclo,
testata prima di ripetere il ciclo e
modificata una volta eseguito il ciclo.



CICLO DI RIPETIZIONE SU CONTATORE



Stampare la tabellina del 3.



3	x	0	=	0
3	x	1	=	3
3	x	2	=	6
3	x	3	=	9
3	x	4	=	12
3	x	5	=	15
3	x	6	=	18
3	x	7	=	21
3	x	8	=	24
3	x	9	=	27
3	x	10	=	30

PROBLEMA DI INFORMATICA

CICLO DI RIPETIZIONE SU CONDIZIONE

Se non sappiamo a priori **quante** volte dobbiamo eseguire il ciclo, sicuramente sappiamo **quando** eseguirlo e **quando** no.



I passi elementari
vengono eseguiti
più volte

ITERAZIONE

Anche questa volta
ci sono due casi.

Prima eseguiamo
il ciclo e poi facciamo il test?
Oppure prima facciamo il test
e poi eseguiamo il ciclo ?

CICLO DI RIPETIZIONE SU CONDIZIONE

Mangiare un piatto di spaghetti.



PROBLEMA DEL MONDO REALE



CICLO DI RIPETIZIONE SU CONDIZIONE CON CONDIZIONE IN CODA

Mangiare un piatto di spaghetti.



PROBLEMA DEL MONDO REALE

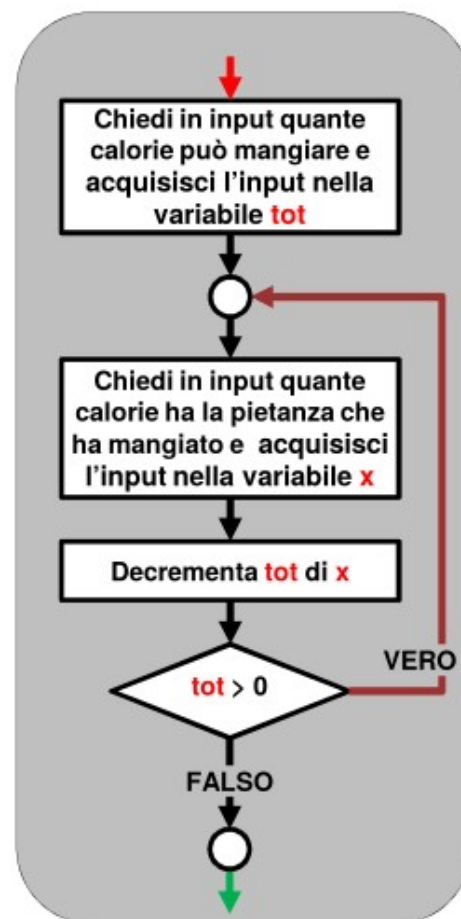


CICLO DI RIPETIZIONE SU CONDIZIONE CON CONDIZIONE IN CODA

Tenere sotto controllo le calorie.



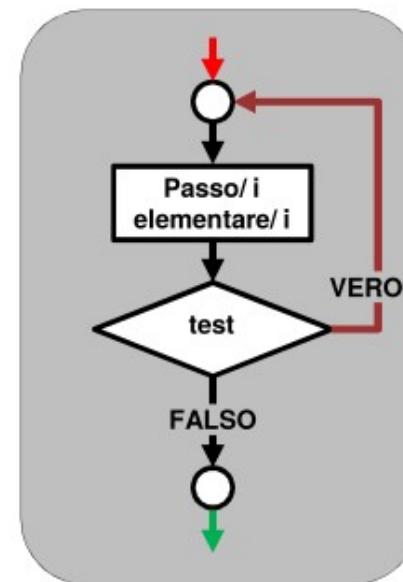
PROBLEMA DI INFORMATICA



CICLO DI RIPETIZIONE SU CONDIZIONE CON CONDIZIONE IN CODA

Il ciclo su condizione con verifica della condizione in coda si presta in due casi:

- ❑ Quando ha senso porsi la domanda **solo dopo** avere già eseguito una prima volta il ciclo (come abbiamo appena visto).
- ❑ Quando c'è un obiettivo da raggiungere mediante uno o più **tentativi** (vediamo un esempio).



CICLO DI RIPETIZIONE SU CONDIZIONE CON CONDIZIONE IN CODA

Chiamare indietro qualcuno.



PROBLEMA DEL MONDO REALE

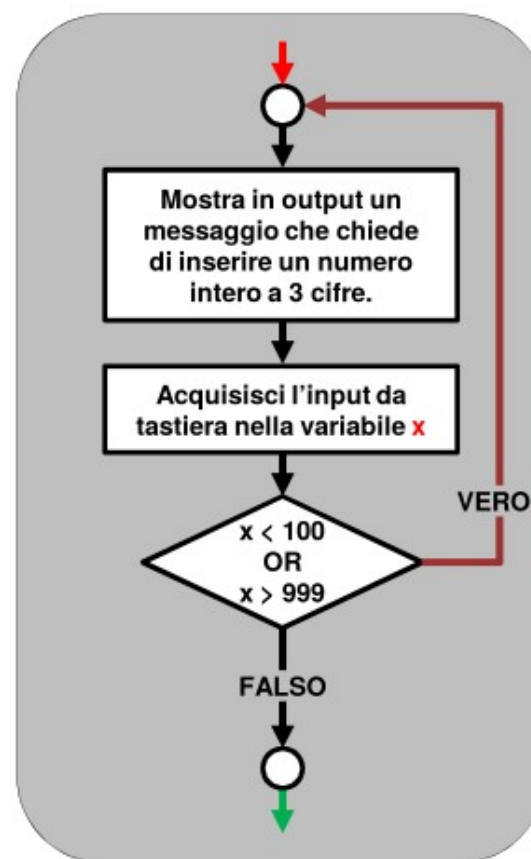


CICLO DI RIPETIZIONE SU CONDIZIONE CON CONDIZIONE IN CODA

Input controllato



PROBLEMA DI INFORMATICA



CICLO DI RIPETIZIONE SU CONDIZIONE

Quando ha senso porsi la domanda **prima**
di eseguire la prima volta il ciclo..



I passi elementari
vengono eseguiti
più volte

ITERAZIONE

..allora si pone
la verifica della
condizione in testa.



Questo tipo di ciclo
si chiama **ciclo di ripetizione
su condizione con verifica
della condizione in testa.**

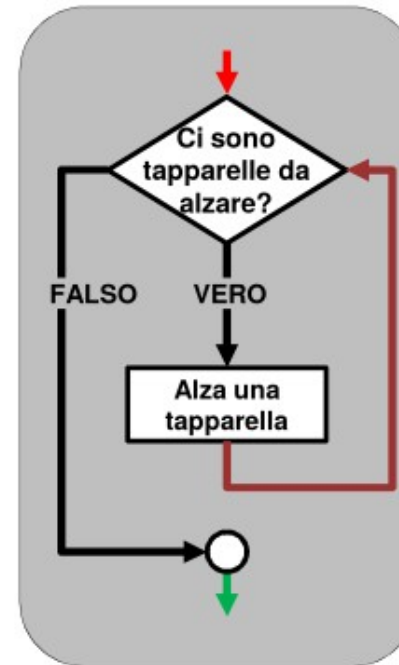


CICLO DI RIPETIZIONE SU CONDIZIONE CON CONDIZIONE IN TESTA

Alzare tutte le tapparelle di casa
(considerando che potrebbero essere già tutte aperte)



PROBLEMA DEL MONDO REALE

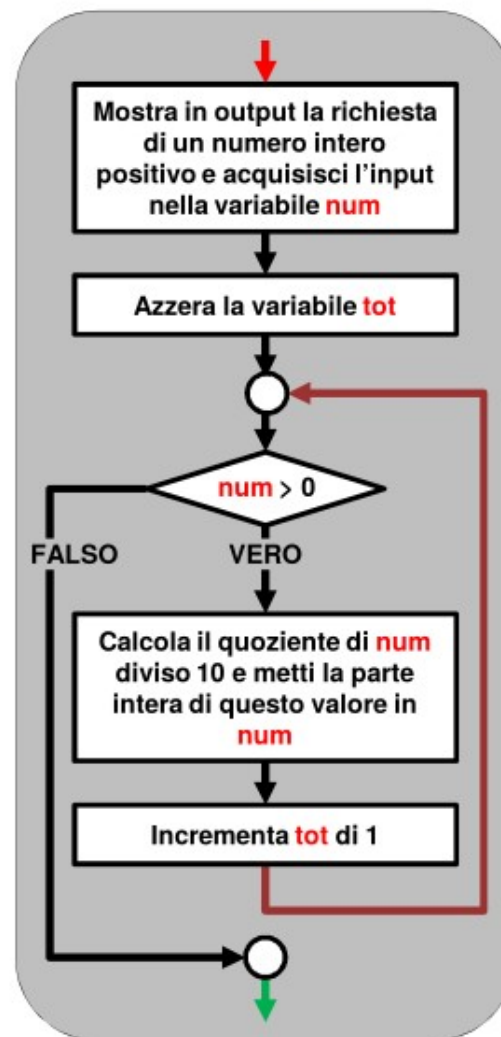


CICLO DI RIPETIZIONE SU CONDIZIONE CON CONDIZIONE IN TESTA

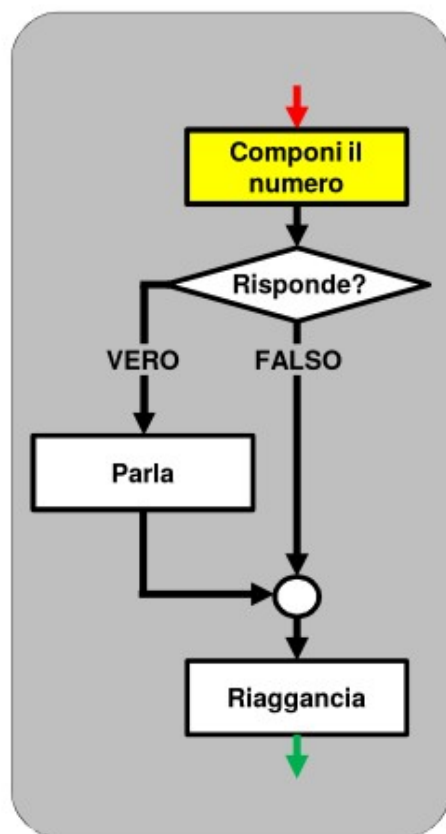
Contare le cifre di un numero



PROBLEMA DI INFORMATICA



COMBINAZIONE DI STRUTTURE PER IL CONTROLLO DEL FLUSSO



Fare una telefonata.

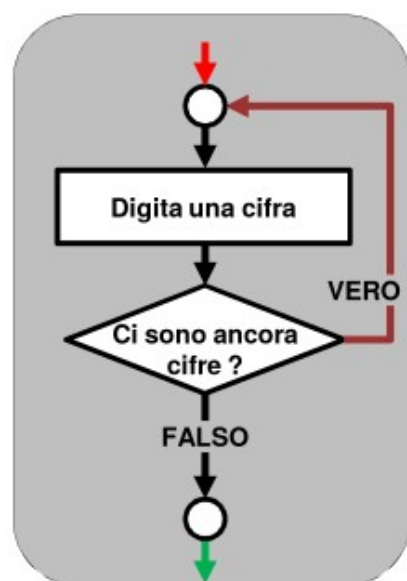


PROBLEMA DEL MONDO REALE

In realtà, il passo "Componi il numero" non è esattamente un passo elementare in quanto è scomponibile.



COMBINAZIONE DI STRUTTURE PER IL CONTROLLO DEL FLUSSO



Comporre un numero.

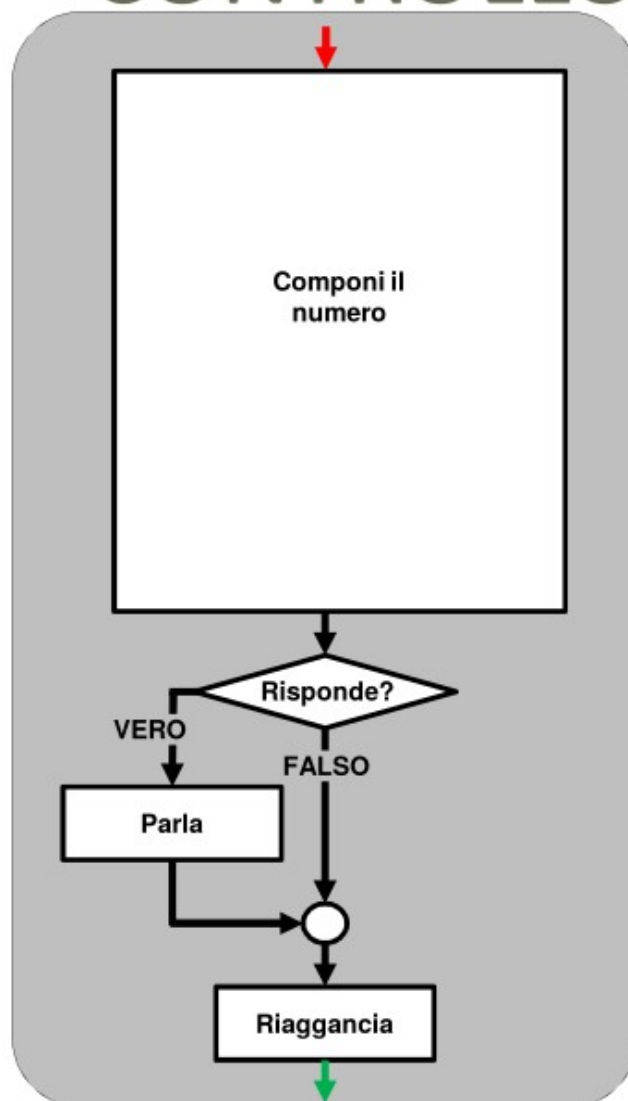


PROBLEMA DEL MONDO REALE

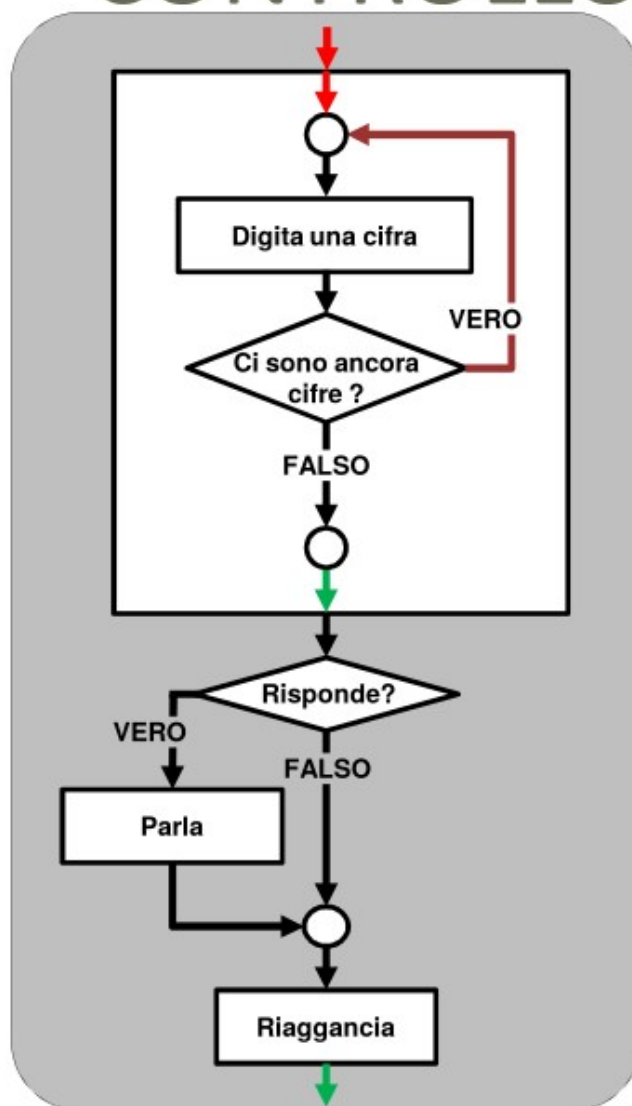
In questo modo l'esercizio precedente diventa..



COMBINAZIONE DI STRUTTURE PER IL CONTROLLO DEL FLUSSO



COMBINAZIONE DI STRUTTURE PER IL CONTROLLO DEL FLUSSO



Fare una telefonata.



PROBLEMA DEL MONDO REALE

Le strutture di controllo possono essere combinate tra loro: questo principio è alla base dell'approccio **top-down**.

SVILUPPO TOP-DOWN

Per sviluppo **top-down** si intende un approccio alla soluzione di un problema che prevede la scomposizione dello stesso in sotto-problemi (più piccoli e quindi più semplici da risolvere).

SVILUPPO TOP-DOWN


Le strutture di controllo possono essere combinate tra loro: questo principio è alla base dell'approccio **top-down**.




SVILUPPO TOP-DOWN


Per sviluppo **top-down** si intende un approccio alla soluzione di un problema che prevede la scomposizione dello stesso in sotto-problemi (più piccoli e quindi più semplici da risolvere).

SVILUPPO TOP-DOWN


Evidentemente  ogni **sotto-problema** sarà risolto con la realizzazione di un **sotto-programma**.

Evidentemente  uno dei passi elementari del nostro algoritmo potrebbe essere la richiesta di esecuzione di uno dei **sotto-programmi**.


SVILUPPO TOP-DOWN




Lo svolgimento di un compito ben preciso (sebbene piccolo) può essere svolto in autonomia, ma più spesso necessita di scambiare dei dati con chi si serve del sotto-programma.



Per esempio, la composizione di un numero richiede che si fornisca il numero da comporre!



I dati che un sotto-programma riceve in input da chi lo invoca si chiamano **parametri**.



Il dato che un sotto-programma fornisce in output può essere chiamato **valore restituito**.

ESEMPIO

1. passo elementare
2. passo elementare
3. passo elementare
4. passo elementare
5. passo elementare
6. passo elementare
7. passo elementare
8. passo elementare
9. passo elementare
10. passo elementare
11. passo elementare
12. passo elementare

1859. passo elementare
1860. passo elementare
1861. passo elementare
1862. passo elementare
1863. passo elementare
1864. passo elementare
1865. passo elementare

Supponiamo di avere un programma
(che questo sia il programma principale o un
sotto-programma non ha alcuna importanza)



Supponiamo poi di avere un
sotto-programma qualsiasi.



1. **passo elementare**
2. passo elementare
3. passo elementare
4. passo elementare
5. passo elementare

127. passo elementare
128. passo elementare
129. passo elementare
130. passo elementare
131. **passo elementare**

ESEMPIO

1. passo elementare
2. passo elementare
3. passo elementare
4. passo elementare
5. passo elementare
6. passo elementare
7. passo elementare
8. passo elementare
9. passo elementare
10. passo elementare
11. passo elementare
12. passo elementare

1859. passo elementare
1860. passo elementare
1861. passo elementare
1862. passo elementare
1863. passo elementare
1864. passo elementare
1865. passo elementare

Quando progettiamo un sotto-programma dobbiamo – tra l'altro – indicare i parametri di cui necessita (**parametri formali**).

Supponiamo che esso si riferisca a questi parametri con i nomi a, b, c, d.

a
b
c
d

1. passo elementare
2. passo elementare
3. passo elementare
4. passo elementare
5. passo elementare

127. passo elementare
128. passo elementare
129. passo elementare
130. passo elementare
131. passo elementare

ESEMPIO

1. passo elementare
2. passo elementare
3. passo elementare
4. passo elementare
5. passo elementare
6. passo elementare
7. passo elementare
8. passo elementare
9. passo elementare
10. passo elementare
11. passo elementare

139. Invoca il sotto-programma qui a lato con i seguenti parametri: n1, n2, n3, n4 e metti il valore restituito nella variabile x.

1859. passo elementare
1860. passo elementare
1861. passo elementare
1862. passo elementare
1863. passo elementare
1864. passo elementare
1865. passo elementare

L'invocazione (da parte del programma chiamante) avviene indicando il nome del sotto-programma, fornendo i parametri (**parametri attuali**) con cui lavorare ed indicando cosa fare del valore restituito.



1. passo elementare
2. passo elementare
3. passo elementare
4. passo elementare
5. passo elementare

127. passo elementare
128. passo elementare
129. passo elementare
130. passo elementare
131. passo elementare

ESEMPIO

1. passo elementare
2. passo elementare
3. passo elementare
4. passo elementare
5. passo elementare
6. passo elementare
7. passo elementare
8. passo elementare
9. passo elementare
10. passo elementare
11. passo elementare

139. Invoca il sotto-programma qui a lato con i seguenti parametri: n1, n2, n3, n4 e metti il valore restituito nella variabile x.

1859. passo elementare
1860. passo elementare
1861. passo elementare
1862. passo elementare
1863. passo elementare
1864. passo elementare
1865. passo elementare

Osserviamo che i nomi di parametri attuali e formali **possono** anche **non** coincidere.

Deve **assolutamente** coincidere il tipo di ogni parametro. E l'ordine.

a
b
c
d

1. passo elementare
2. passo elementare
3. passo elementare
4. passo elementare
5. passo elementare

127. passo elementare
128. passo elementare
129. passo elementare
130. passo elementare
131. passo elementare

ESEMPIO

Il controllo del flusso
passa così al sotto-programma..



1. passo elementare
2. passo elementare
3. passo elementare
4. passo elementare
5. passo elementare
6. passo elementare
7. passo elementare
8. passo elementare
9. passo elementare
10. passo elementare
11. passo elementare

139. Invoca il sotto-programma qui a lato con i seguenti parametri: n1, n2, n3, n4 e metti il valore restituito nella variabile x.

1859. passo elementare
1860. passo elementare
1861. passo elementare
1862. passo elementare
1863. passo elementare
1864. passo elementare
1865. passo elementare

n1
n2
n3
n4

a
b
c
d

1. passo elementare
2. passo elementare
3. passo elementare
4. passo elementare
5. passo elementare

127. passo elementare
128. passo elementare
129. passo elementare
130. passo elementare
131. passo elementare

ESEMPIO

..nel cui parametro a è stato
copiato il valore del parametro n1 e così via.



1. passo elementare
2. passo elementare
3. passo elementare
4. passo elementare
5. passo elementare
6. passo elementare
7. passo elementare
8. passo elementare
9. passo elementare
10. passo elementare
11. passo elementare

139. Invoca il sotto-programma qui
a lato con i seguenti parametri:
n1, n2, n3, n4 e metti il valore
restituito nella variabile x.

1859. passo elementare
1860. passo elementare
1861. passo elementare
1862. passo elementare
1863. passo elementare
1864. passo elementare
1865. passo elementare

n1
n2
n3
n4

a
b
c
d

1. passo elementare
2. passo elementare
3. passo elementare
4. passo elementare
5. passo elementare

127. passo elementare
128. passo elementare
129. passo elementare
130. passo elementare
131. passo elementare

ESEMPIO



Quando il sotto-programma ha eseguito la sua ultima istruzione, questi fornisce al programma chiamante il suo **valore di restituzione** e l'esecuzione riprende dall'istruzione successiva (del chiamante).

1. passo elementare
2. passo elementare
3. passo elementare
4. passo elementare
5. passo elementare
6. passo elementare
7. passo elementare
8. passo elementare
9. passo elementare
10. passo elementare
11. passo elementare

139. Invoca il sotto-programma qui a lato con i seguenti parametri: n1, n2, n3, n4 e metti il valore restituito nella variabile x.

1859. passo elementare
1860. passo elementare
1861. passo elementare
1862. passo elementare
1863. passo elementare
1864. passo elementare
1865. passo elementare

n1
n2
n3
n4

x

a
b
c
d

1. passo elementare
2. passo elementare
3. passo elementare
4. passo elementare
5. passo elementare

127. passo elementare
128. passo elementare
129. passo elementare
130. passo elementare
131. passo elementare

ISTRUZIONI

! Riassumendo,
ogni passo elementare
sarà una istruzione del seguente elenco.

- ☐ di input
- ☐ di output
- ☐ di assegnazione
- ☐ di selezione
- ☐ di iterazione
- ☐ di invocazione di funzione

ISTRUZIONI

