

ELEMENTI BASE ALGORITMI

Variabili, Definizioni, Assegnamenti, Selezione, cicli for e while

VARIABILI

Una variabile è un contenitore di dati destinato a contenere valori di diverso tipo, il valore contenuto al suo interno può essere modificato durante l'esecuzione di un programma. Una variabile è caratterizzata da un nome e da una definizione di tipo.

TIPI PRINCIPALI:

- Integer: numeri interi
- Real: numeri reali
- String: testo
- Boolean: vero, falso

ESEMPIO:

String FRASE

FRASE



DEFINIZIONE

Le variabili devono essere definite (dichiarate) in maniera tipizzata prima di essere adoperate.

TIPI PRINCIPALI:

- Integer: numeri interi
- Real: numeri reali
- String: testo
- Boolean: vero, falso

ESEMPIO:

String FRASE

FRASE



ASSEGNAMENTO

Le operazioni basilari che possono essere compiute su una variabile sono la lettura del valore in essa contenuto e la scrittura di un nuovo valore, o assegnamento.

Il primo assegnamento della variabile è detto inizializzazione.

TIPI PRINCIPALI:

- Integer: numeri interi
- Real: numeri reali
- String: testo
- Boolean: vero, falso

ESEMPIO:

String FRASE = "ciao"

FRASE

ciao

ESEMPIO DICHIARAZIONE E ASSEGNAIMENTO

INT X;	DICHIARAZIONE VARIABILE X DI TIPO INTERO
X = 10;	INIZIALIZZAZIONE DELLA VARIABILE X CON IL VALORE 10
X = X + 1;	LETTURA DEL VALORE CONTENUTO IN X; (10) ESECUZIONE DELL'ISTRUZIONE X + 1; (10 + 1) ASSEGNAIMENTO DEL VALORE RISULTANTE ALLA VARIABILE X (11)
SCRIVI X;	ESECUZIONE DELL'ISTRUZIONE DI STAMPA DEL VALORE CONTENUTO IN X (11)

OPERAZIONI

Tra valori e variabili possono essere eseguite diverse operazioni, le operazioni principalmente utilizzate sono la somma, la sottrazione, il prodotto, la divisione, il maggiore, il minore, l'uguale, il diverso, il modulo, gli operatori logici booleani e il concatenamento tra stringhe.

ESEMPIO `INT A = 10; INT B = 4`

+	SOMMA	A + B risultato 14
-	SOTTRAZIONE	A - B risultato 6
*	PRODOTTO	A * B risultato 40
/	DIVISIONE	A / B risultato 2
%	MODULO (resto)	A % B risultato 2
>	MAGGIORE	A > B risultato VERO
<	MINORE	A < B risultato FALSO
=	UGUALE	A = B risultato FALSO
!=	DIVERSO	A != B risultato VERO
AND	AND LOGICO	
OR	OR LOGICO	
&	CONCATENAMENTO	

AND, OR, CONCATENAMENTO (INT A = 10; INT B = 4)

A	B	A AND B
VERO	VERO	VERO
VERO	FALSO	FALSO
FALSO	VERO	FALSO
FALSO	FALSO	FALSO

A	B	A OR B
VERO	VERO	VERO
VERO	FALSO	VERO
FALSO	VERO	VERO
FALSO	FALSO	FALSO

ESEMPIO AND: $A < 5$ AND $B = 4$; **FALSO** AND **VERO**; risultato: **FALSO**

ESEMPIO AND: $A = 10$ AND $B = 4$; **VERO** AND **VERO**; risultato: **VERO**

ESEMPIO OR: $A < 5$ OR $B = 4$; **FALSO** OR **VERO**; risultato: **VERO**

ESEMPIO OR: $A = 10$ OR $B = 4$; **VERO** OR **VERO**; risultato: **VERO**

CONCATENAMENTO (STRING TXT = "RISULTATO"; INT X = 42)

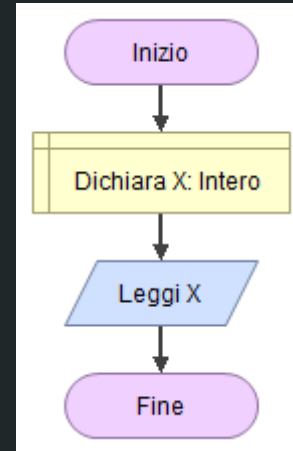
TXT & X	RISULTATO42
TXT & " " & X	RISULTATO 42
"il " & TXT & " è: " & X	il RISULTATO è: 42

INPUT/OUTPUT

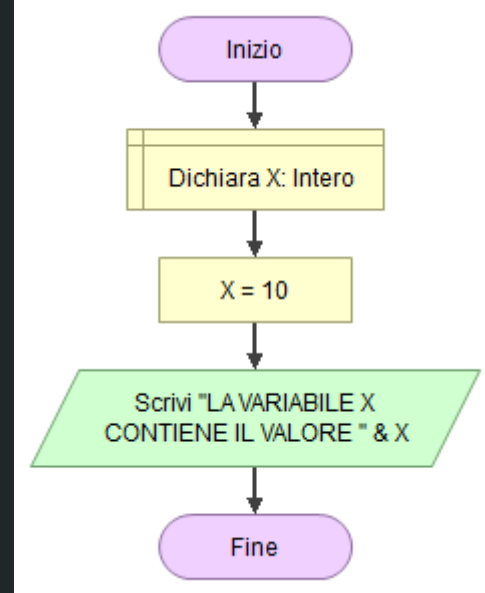
Per leggere un valore in **INPUT** si utilizza l'istruzione **LEGGI** che blocca l'esecuzione del programma in attesa di un valore che deve essere inserito dall'utente. Una volta letto il valore esso viene assegnato a una variabile definita in precedenza nel programma.

Per stampare un valore in **OUTPUT** si utilizza l'istruzione **SCRIVI** che stampa a video il valore desiderato.

INPUT



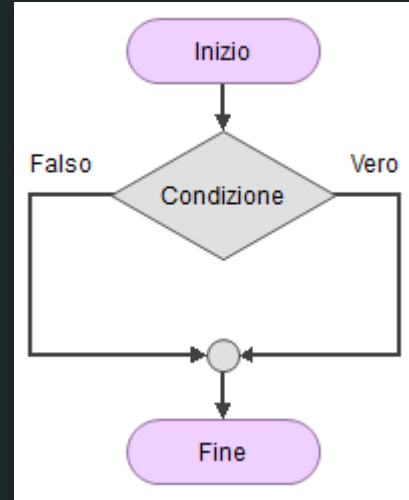
OUTPUT



COSTRUTTO DI SELEZIONE (IF)

SELEZIONE

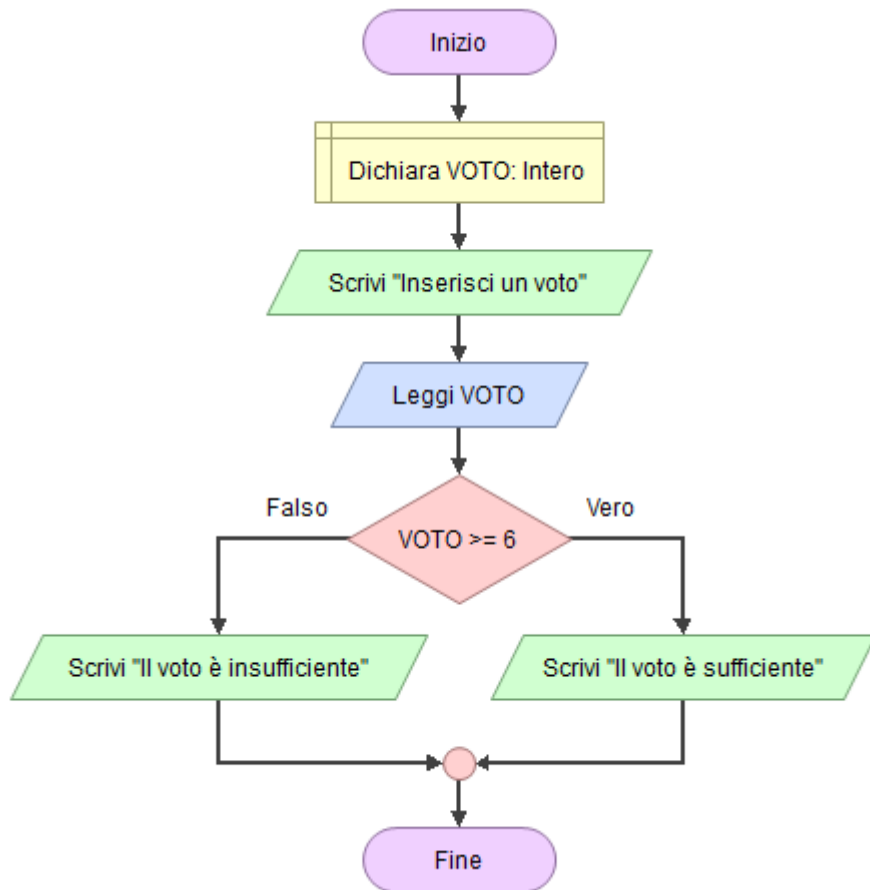
La selezione è una struttura di controllo del flusso di esecuzione di un programma che indica all'elaboratore, in base alla verifica di una condizione logica specificata, quale fra due sequenze o blocchi di istruzioni eseguire.



ESERCIZI SVOLTI

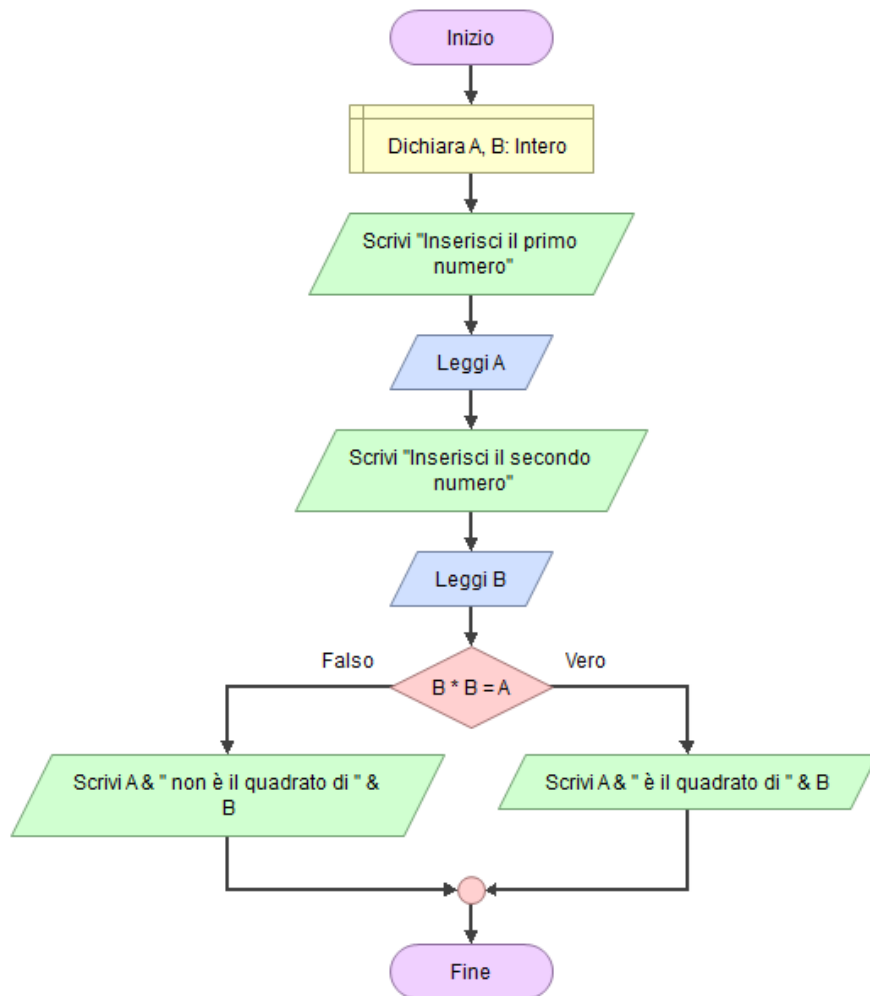
ESERCIZIO 4

Determina se un voto ricevuto in ingresso è sufficiente o insufficiente.



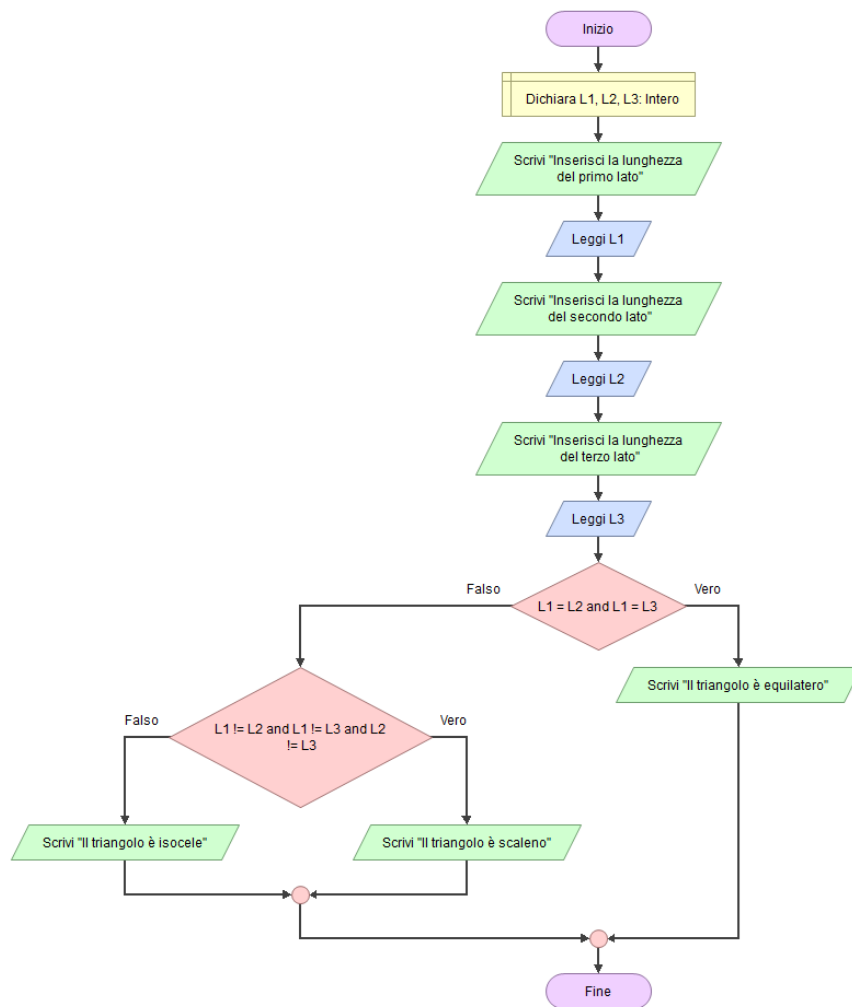
ESERCIZIO 6

Dati due numeri A e B verificare se A è il quadrato di B



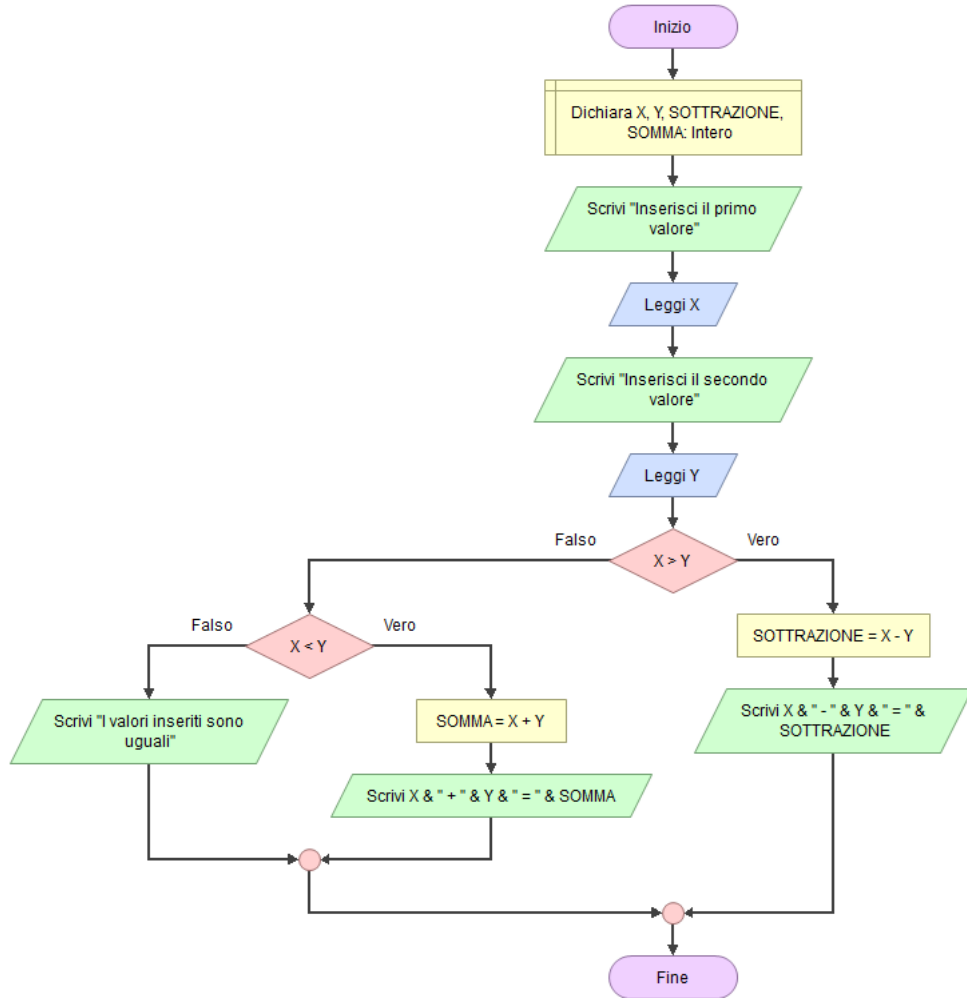
ESERCIZIO 2

Ricevute in ingresso le lunghezze dei tre lati di un triangolo, determina se si tratta di un triangolo equilatero, isoscele, scaleno.



ESERCIZIO 3

Ricevuti in ingresso due valori, fanno la sottrazione se il primo è maggiore del secondo o fanno la somma se il primo è minore del secondo. Se sono uguali segnalalo a video.

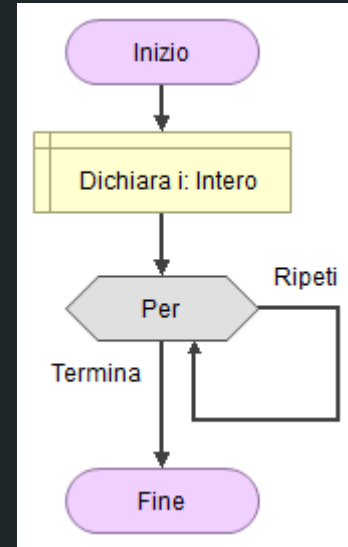


CICLO FOR (PER)

CICLO FOR

Il ciclo FOR viene principalmente utilizzato quando si vogliono ripetere delle istruzioni un determinato numero di volte.

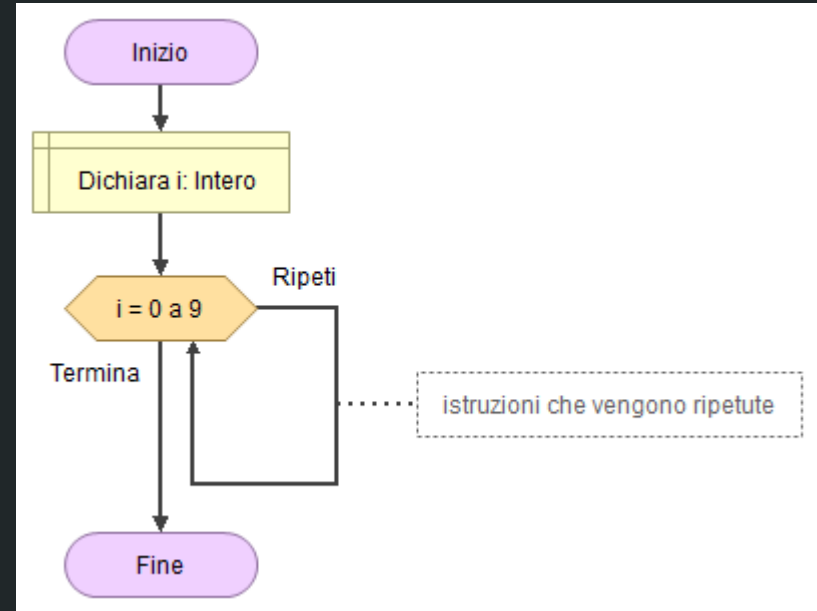
in genere si è quindi in grado di stabilire per quante volte verrà ripetuto il ciclo prima che il programma venga eseguito.



CICLO FOR

Per realizzare il ciclo in genere viene utilizzata una variabile *i*, inizializzata ad un valore di partenza, che viene incrementata ad ogni passo, fino a che non raggiunge un valore finale. Nel momento in cui la variabile *i* raggiunge il valore finale, il ciclo termina e l'esecuzione del programma procede dall'istruzione successiva al ciclo.

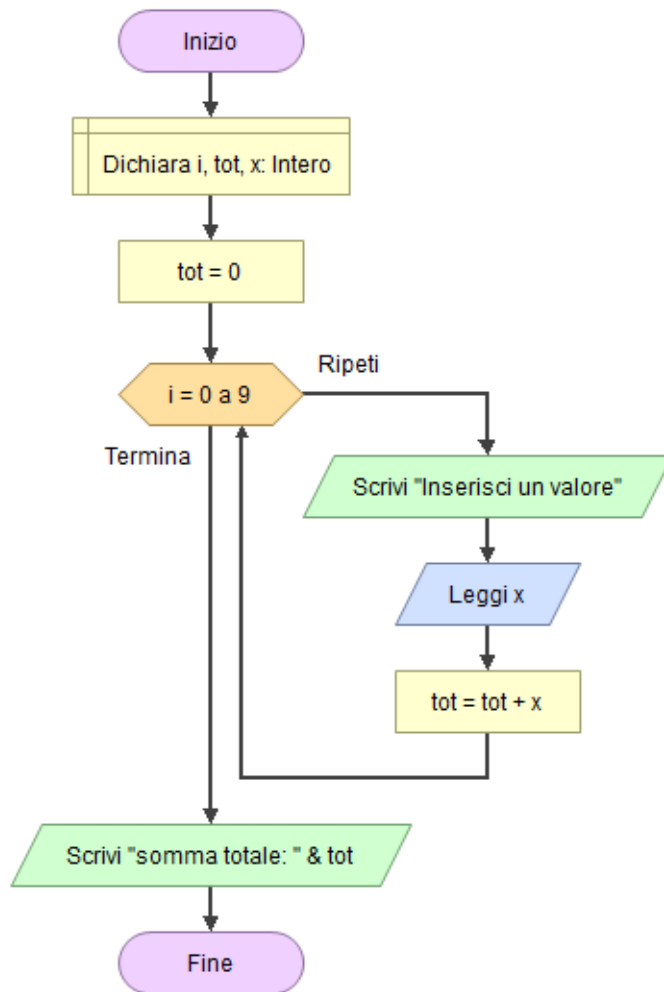
ESEMPIO: ciclo ripetuto 10 volte



ESERCIZI SVOLTI

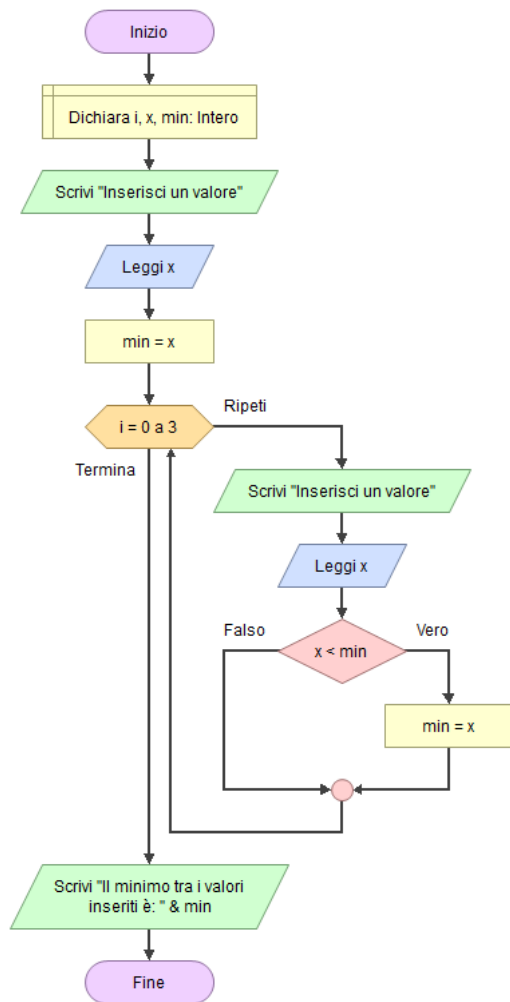
ESERCIZIO 10

Leggere in input 10 numeri e calcolarne la somma.



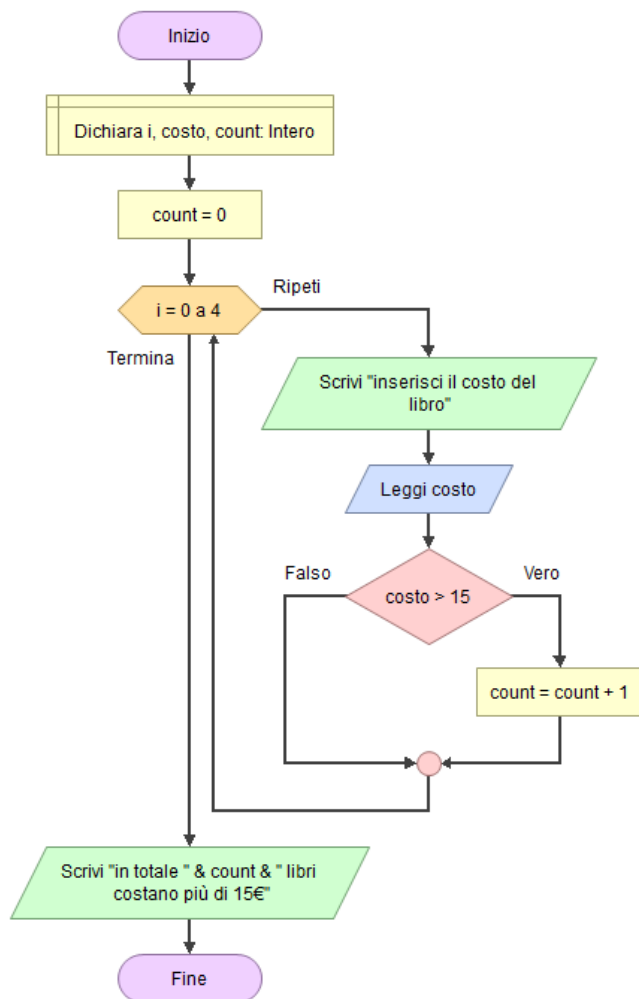
ESERCIZIO 12

Scrivere un programma che legge 5 numeri da tastiera e ne restituisce il minimo.



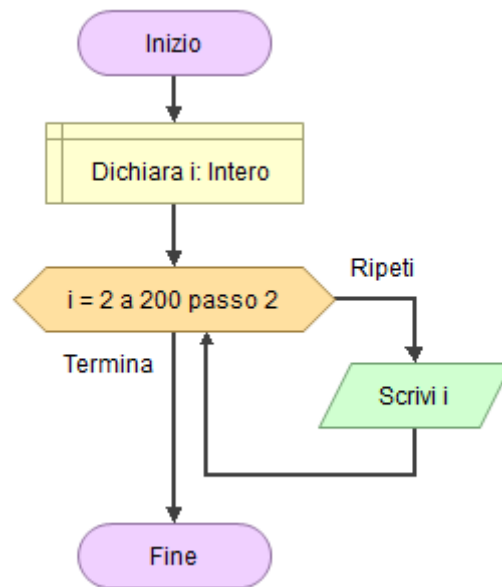
ESERCIZIO 13

Dato in ingresso il costo di cinque libri,
conta quanti di questi hanno un prezzo
superiore a 15 euro.



ESERCIZIO 15

Visualizza i primi 100 numeri pari
partendo da 2.

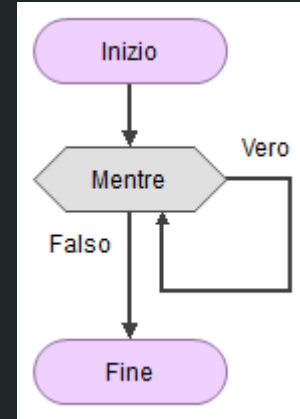


CICLO WHILE (MENTRE)

CICLO WHILE

Il ciclo WHILE viene principalmente utilizzato quando si vogliono ripetere delle istruzioni un indeterminato numero di volte.

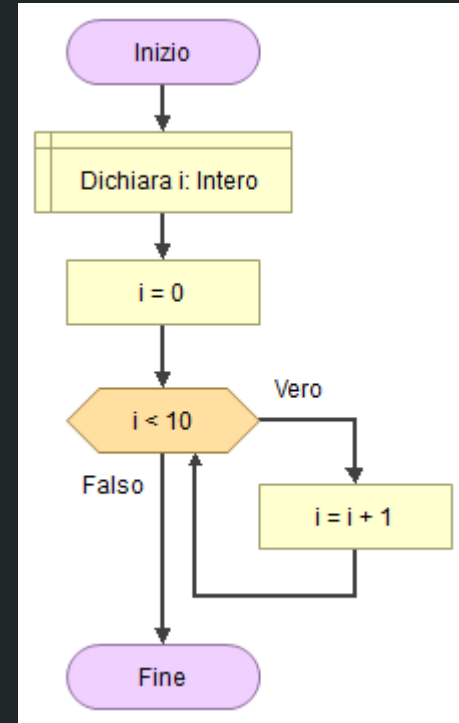
in genere NON si è quindi in grado di stabilire per quante volte verrà ripetuto il ciclo prima che il programma venga eseguito.



CICLO WHILE

Ad ogni passo viene controllata la validità della condizione del ciclo while, se la condizione è verificata le istruzioni contenute nel ciclo vengono ripetute, se non è verificata il ciclo termina e l'esecuzione del programma procede dall'istruzione successiva al ciclo.

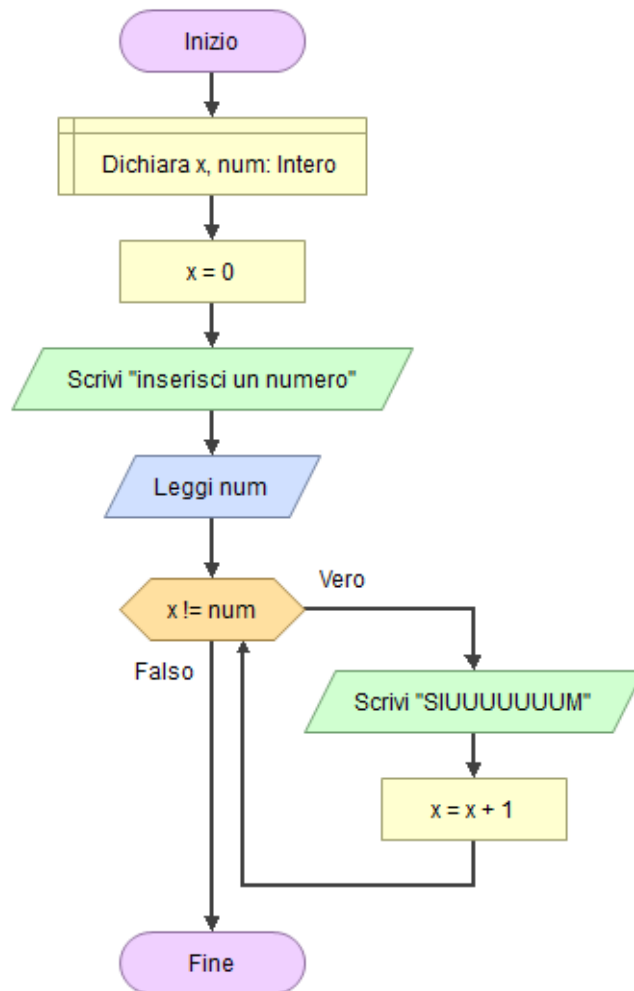
ESEMPIO: ciclo ripetuto 10 volte



ESERCIZI SVOLTI

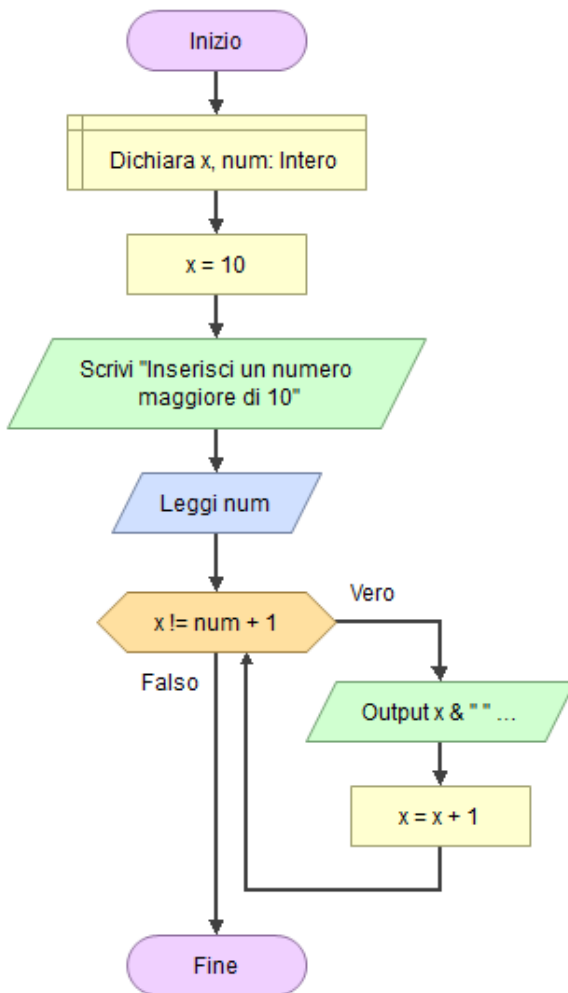
ESERCIZIO 16

Dato un valore intero inserito dall'utente stampa la frase "SIUUUUUUUUUUM" tante volte quante il numero inserito dall'utente.



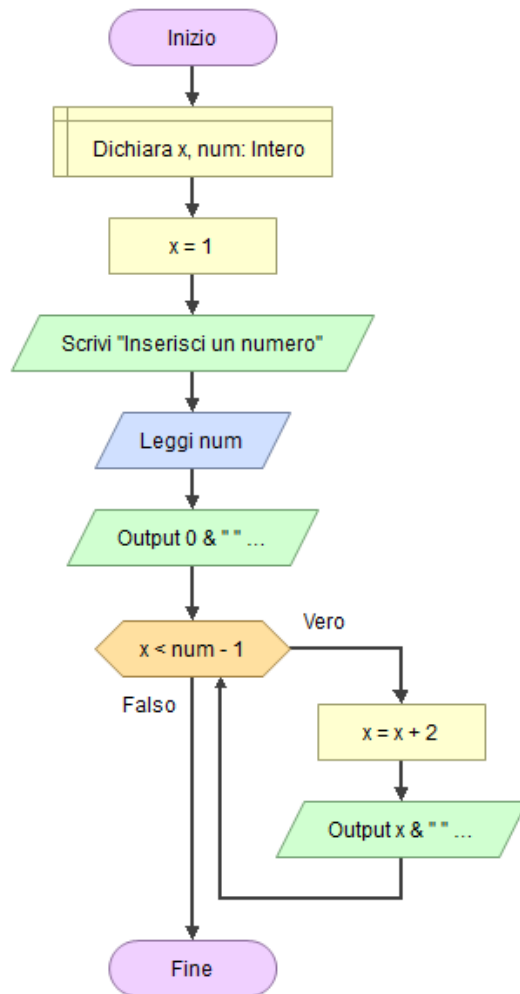
ESERCIZIO 19

Scrivi un algoritmo che richieda in ingresso un numero num maggiore di 10 e stampi tutti i valori interi compresi tra 10 e num.



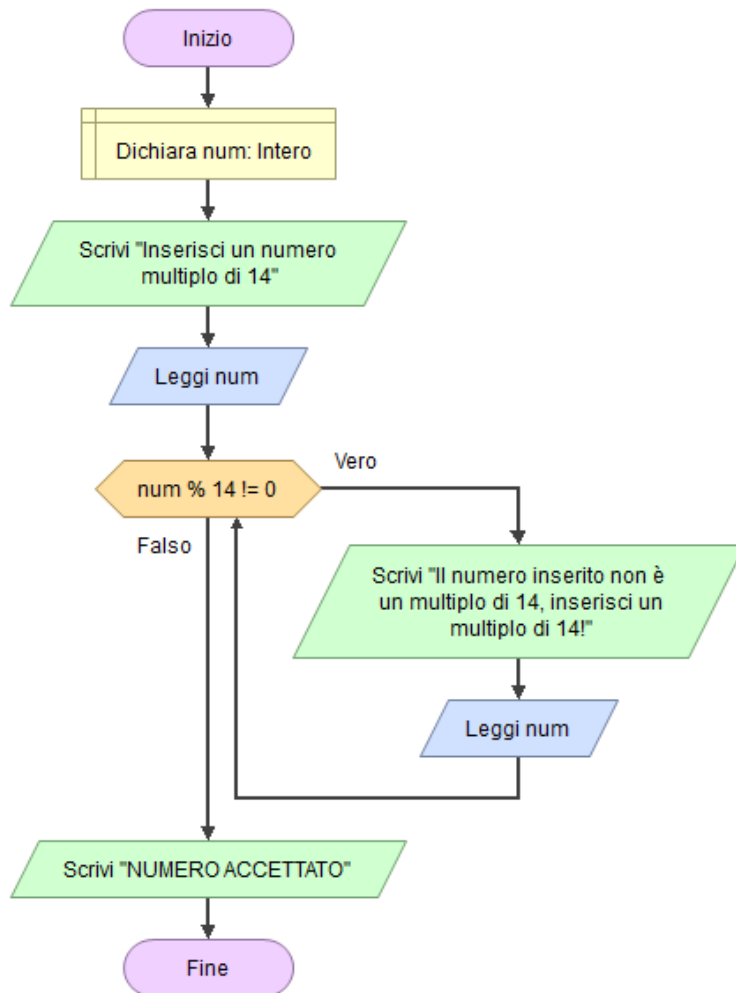
ESERCIZIO 20

Scrivi un algoritmo che richieda in ingresso un numero num e stampi tutti i valori interi dispari compresi tra 0 e num.



ESERCIZIO 21

Scrivi un algoritmo che richieda in ingresso un numero num, accettandolo solo se è un multiplo di 14.



FINE
