

# Fondamenti di Informatica

Prof. Ing. Loris Penserini, PhD

[elpense@gmail.com](mailto:elpense@gmail.com)

<https://orcid.org/0009-0008-6157-0396>

Materiale:

[https://github.com/penserini/Lezioni UnivPM.git](https://github.com/penserini/Lezioni_UnivPM.git)



# **Realizzare Algoritmi con procedimento iterativo**



# Project Work

## **Problema (pari/dispari)**

Realizzare un algoritmo che dato in input un qualsiasi numero intero positivo verifica se è dispari o pari e ne comunichi il risultato.



# PW: Soluzione informale

## Problema (pari/dispari)

Realizzare un algoritmo che dato un numero intero positivo verifica se è dispari o pari e ne comunichi il risultato.

## Idea (informale)

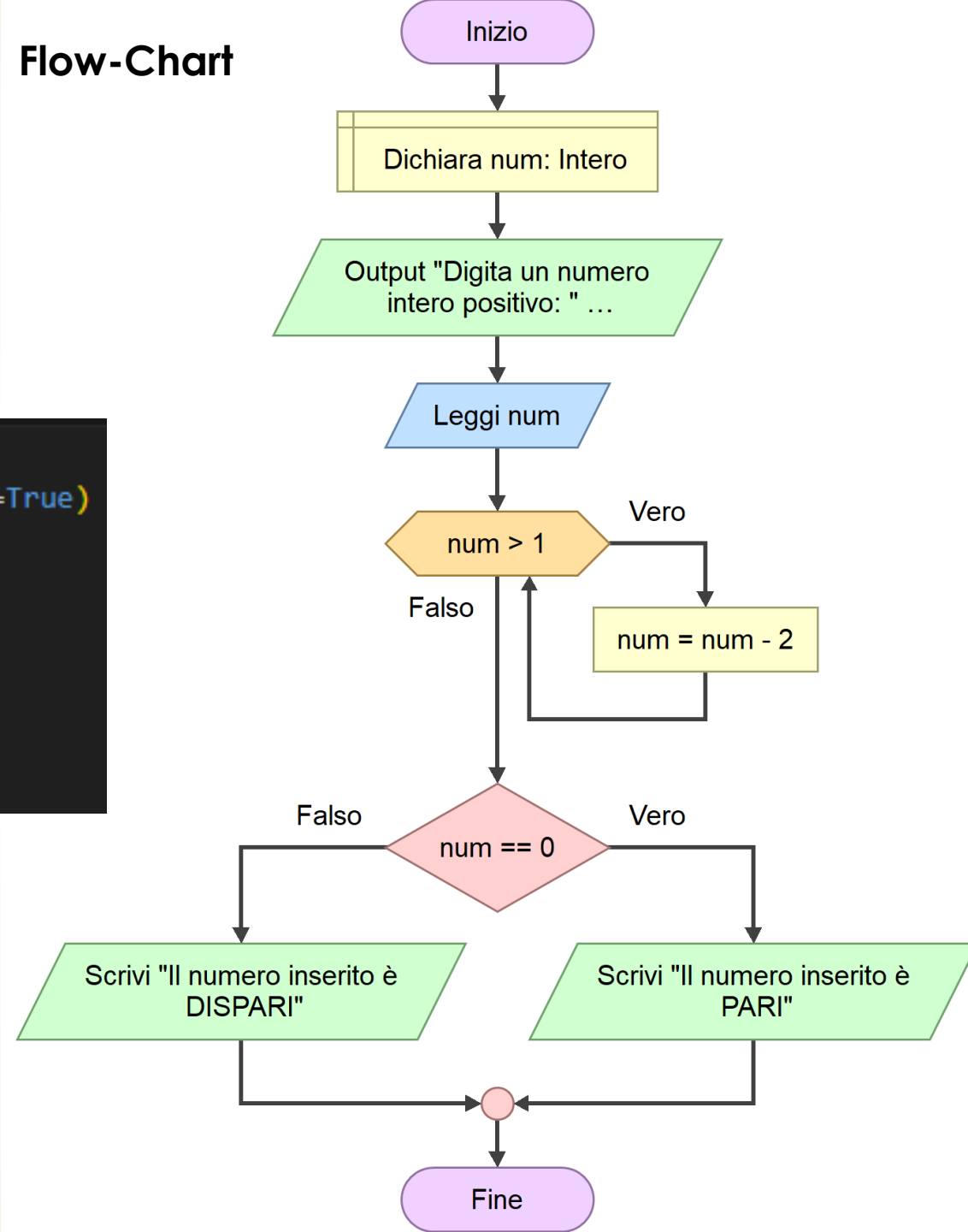
Un numero pari è divisibile per 2, per cui tolgo 2 al numero inserito tante volte fino a quando ho resto 1 o 0. Se il resto è 0 allora il numero iniziale inserito sarà un numero pari e viceversa.

# PW: Soluzione

## Codice Python

```
❏ pari_dispari.py > ...  
1 print("Digita un numero intero positivo: ", end='', flush=True)  
2 num = int(input())  
3 while num > 1:  
4     num = num - 2  
5 if num == 0:  
6     print("Il numero inserito è PARI")  
7 else:  
8     print("Il numero inserito è DISPARI")
```

## Flow-Chart



# Progetto della soluzione: Java

```
PariDispari.java ✕
1 import java.io.BufferedReader;
4
5 public class PariDispari {
6
7     public static void main(String[] args) throws IOException{
8         int num = 0;
9         String str;
10
11         InputStreamReader input = new InputStreamReader(System.in);
12         BufferedReader tastiera = new BufferedReader(input);
13
14         System.out.print("Digita un numero intero positivo: ");
15         System.out.print("");
16         str = tastiera.readLine();
17         num = Integer.parseInt(str);
18
19         while(num > 1) {
20             num = num - 2;
21         }
22         if(num == 0) {
23             System.out.println("Il numero inserito è PARI");
24         } else {
25             System.out.println("Il numero inserito è DISPARI");
26         }
27     }
28 }
```



# Project Work

## Problema (min)

Realizzare un algoritmo che riceva in input una serie di temperature e produca in output la temperatura minima.

L'inserimento deve continuare fino a che l'utente decida di uscire.



# PW: Soluzione informale

## Problema (min)

Realizzare un algoritmo che riceva in input una serie di temperature e produca in output la temperatura **minima**.

L'inserimento deve continuare fino a che l'utente decida di uscire.

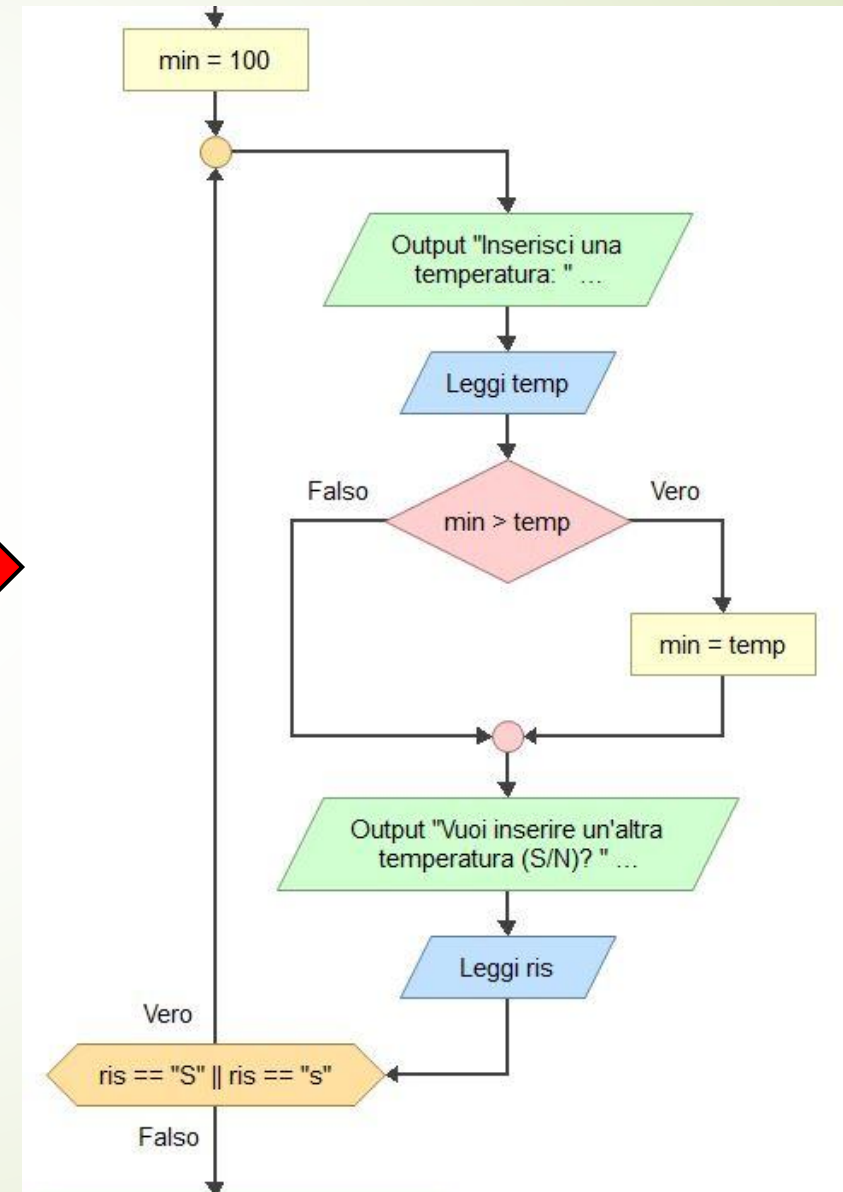
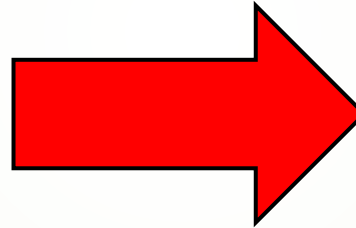
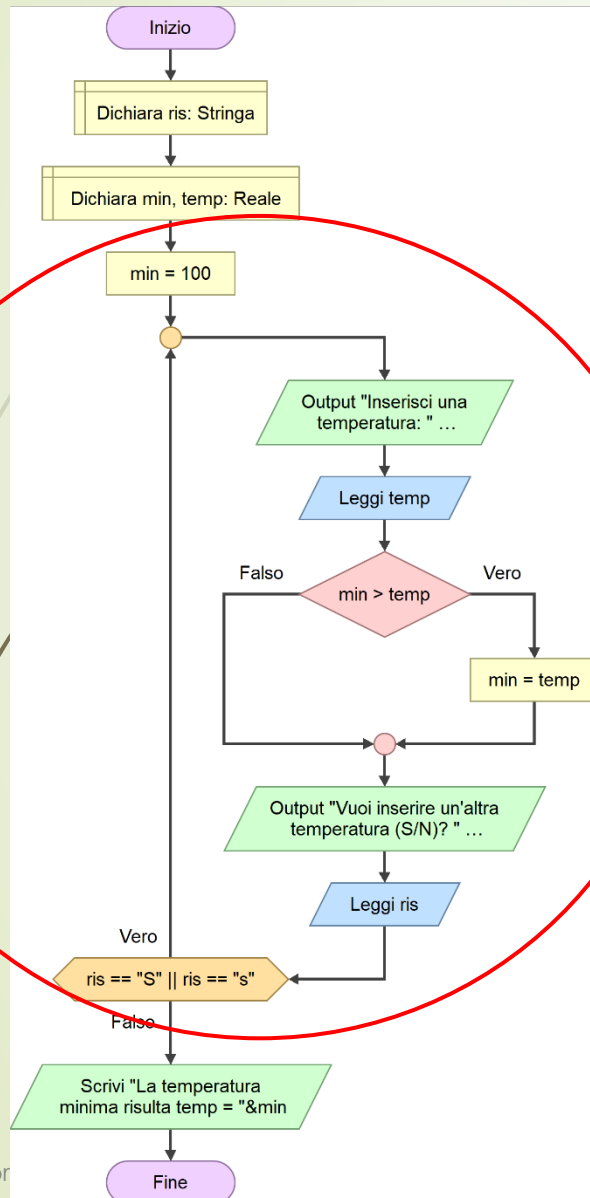
## Idea (informale)

Realizzo l'input dentro un ciclo e, ad ogni inserimento, testo con una domanda per l'utente se vuole continuare ad inserire altre temperature.

Inizializzo una variabile «min» ad un valore iniziale alto, così da non perturbare la ricerca del minimo sull'insieme dato. Così «min» servirà da accumulatore del minimo relativo, operando un confronto tutte le volte che l'utente inserisce una temperatura. Al termine del ciclo, o fase di inserimento, «min» conterrà il minimo assoluto.



# PW: Progetto della soluzione



# PW: Realizzazione in Python

```
minimo_temp.py > ...  
1  min = 100  
2  while True:      #This simulates a Do Loop  
3      print("Inserisci una temperatura °C: ", end='', flush=True)  
4      temp = float(input())  
5      if min > temp:  
6          min = temp  
7      print("Vuoi inserire un'altra temperatura (S/N)? ", end='', flush=True)  
8      ris = input()  
9      if not(ris == "S" or ris == "s"): break    #Exit loop  
10 print("La temperatura minima risulta temp = " + str(min) + " °C")
```

Se gli input sono le temperature di fusione dei metalli, l'algoritmo continua a funzionare correttamente?

# PW: Realizzazione in JAVA

```
Min.java ✖
1 import java.io.*;
2
3 public class Min {
4
5     public static void main(String[] args) throws IOException{
6         double min = 100.00;
7         double temp;
8         String str;
9         String ris = "S";
10
11         InputStreamReader input = new InputStreamReader(System.in);
12         BufferedReader tastiera = new BufferedReader(input);
13
14         do {
15             System.out.print("Inserisci una temperatura: ");
16             str = tastiera.readLine();
17             temp = Double.parseDouble(str);
18             if(min > temp) {
19                 min = temp;
20             }
21             System.out.println("Vuoi inserire un'altra temperatura (S/N)? ");
22             ris = tastiera.readLine();
23
24         } while(ris.equals("S") || ris.equals("s"));
25
26         System.out.println("La temperatura minima risulta temp = " + min);
27     }
28 }
```

# Project Work

## Problema (min e max)

Realizzare un algoritmo che riceva in input una serie di temperature e produca in output la temperatura **minima** e quella **massima**.

L'utente deve fornire in input il numero totale (ntemp) dei valori di temperature da inserire. L'algoritmo deve funzionare per qualsiasi valore di temperatura inserito.

# PW: Soluzione informale

## Problema (min e max)

Realizzare un algoritmo che riceva in input una serie di temperature e produca in output la temperatura minima e quella massima.

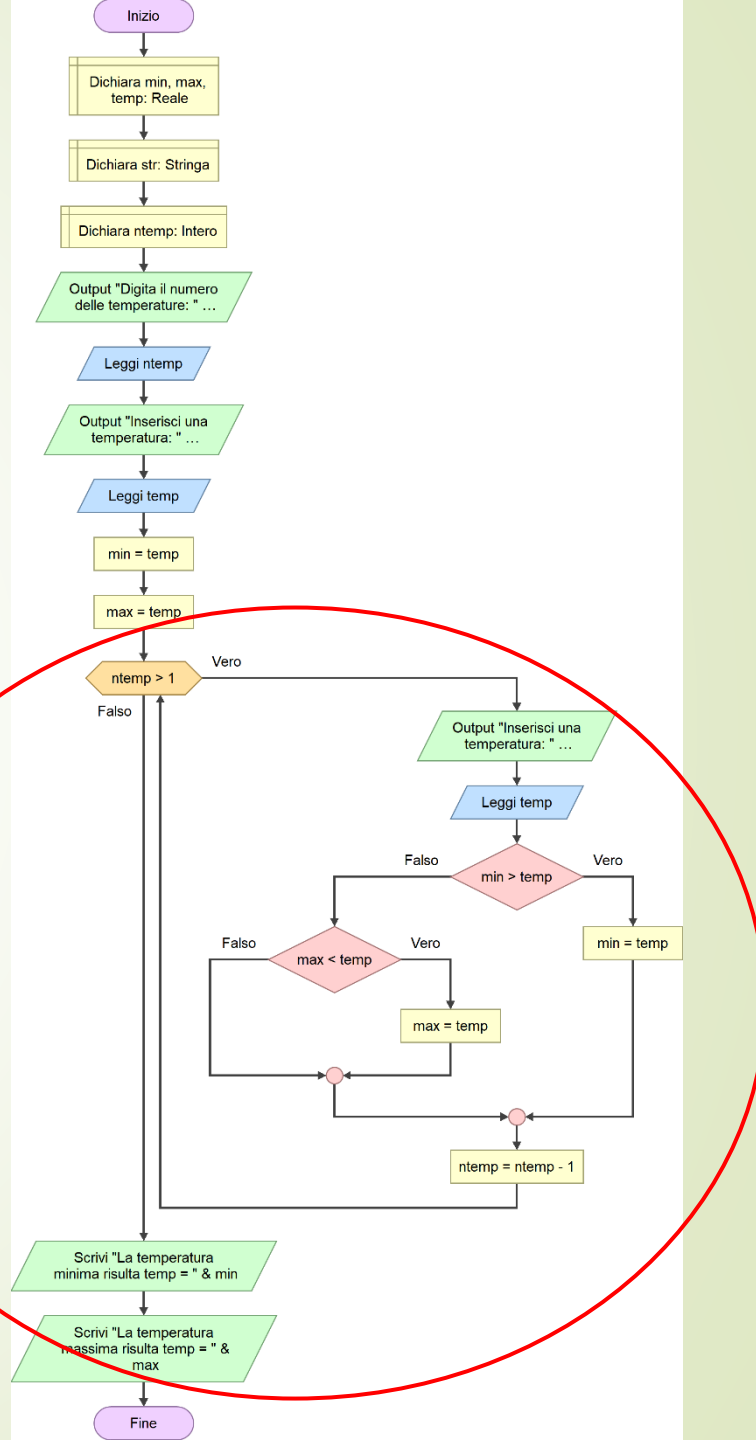
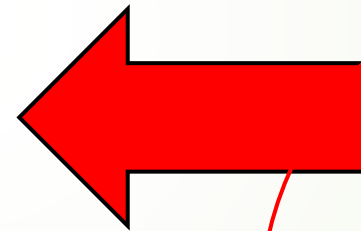
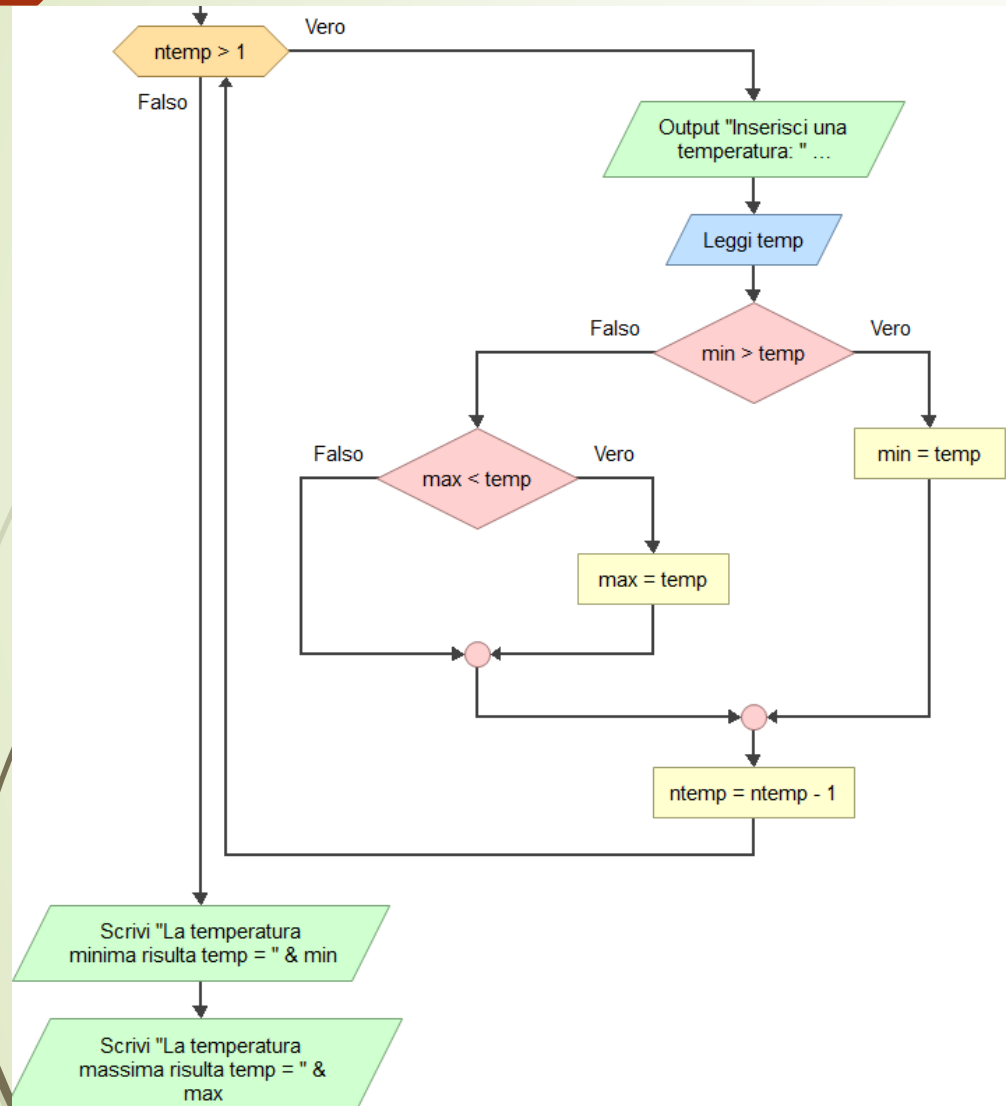
L'utente deve fornire, a priori come input, il numero totale (ntemp) dei valori di temperature da inserire. L'algoritmo deve funzionare per qualsiasi valore di temperatura inserito.

## Idea (informale)

L'input «ntemp» determina il numero dei cicli. Utilizzo il primo inserimento di «temp» per inizializzare sia «min» che «max», che serviranno da accumulatori del minimo e del massimo relativi, operando un confronto tutte le volte che l'utente inserisce una temperatura.

Al termine del ciclo, o fase di inserimento, «min» e «max» conterranno il minimo e il massimo assoluti.

# PW: Progetto della soluzione



# PW: Realizzazione in Python

```
min_max_temp.py > ...
1  print("Digita il numero delle temperature: ", end='', flush=True)
2  ntemp = int(input())
3  print("Inserisci una temperatura °C: ", end='', flush=True)
4  temp = float(input())
5  min = temp
6  max = temp
7  while ntemp > 1:
8      print("Inserisci una temperatura °C: ", end='', flush=True)
9      temp = float(input())
10     if min > temp:
11         min = temp
12     else:
13         if max < temp:
14             max = temp
15     ntemp = ntemp - 1
16 print("La temperatura minima risulta temp = ", min, "°C")
17 print("La temperatura massima risulta temp = " + str(max) + " °C")
```



# PW: Realizzazione in Java

```
MinMax.java
1 import java.io.*;
2
3 public class MinMax {
4
5     public static void main(String[] args) throws IOException{
6         double min,max;
7         double temp;
8         int ntemp = 0;
9         String str;
10
11         InputStreamReader input = new InputStreamReader(System.in);
12         BufferedReader tastiera = new BufferedReader(input);
13
14         System.out.print("Digita il numero di temperature: ");
15         str = tastiera.readLine();
16         ntemp = Integer.parseInt(str);
17
18         System.out.print("");
19         System.out.print("Inserisci una temperatura: ");
20         str = tastiera.readLine();
21         min = Double.parseDouble(str);
22         max = min;
23
24         while(ntemp > 1) {
25             System.out.print("Inserisci una temperatura: ");
26             str = tastiera.readLine();
27             temp = Double.parseDouble(str);
28             if(min > temp) {
29                 min = temp;
30             } else if(max < temp) {
31                 max = temp;
32             }
33             ntemp = ntemp -1;
34         }
35         System.out.println("La temperatura minima risulta temp = " + min);
36         System.out.println("La temperatura massima risulta temp = " + max);
37     }
38 }
```

# Project Work

## Problema (potenza)

Realizzare un algoritmo che riceva in input un numero reale, la «base», e un numero intero positivo, l' «esponente», e calcoli e stampi l'elevamento a potenza. Ovviamente senza utilizzare eventuali operatori già preconfezionati del linguaggio adottato. Per esempio:

**base = 3**

**esponente = 2**

**=>**

**$3^2 = 9$**

# PW: Soluzione informale

## Problema (potenza)

Realizzare un algoritmo che riceva in input un numero reale, la «base», e un numero intero positivo, l' «esponente», e calcoli e stampi l'elevamento a potenza. Ovviamente senza utilizzare eventuali operatori già preconfezionati del linguaggio adottato. Per esempio:

**base = 3**

**esponente = 2**

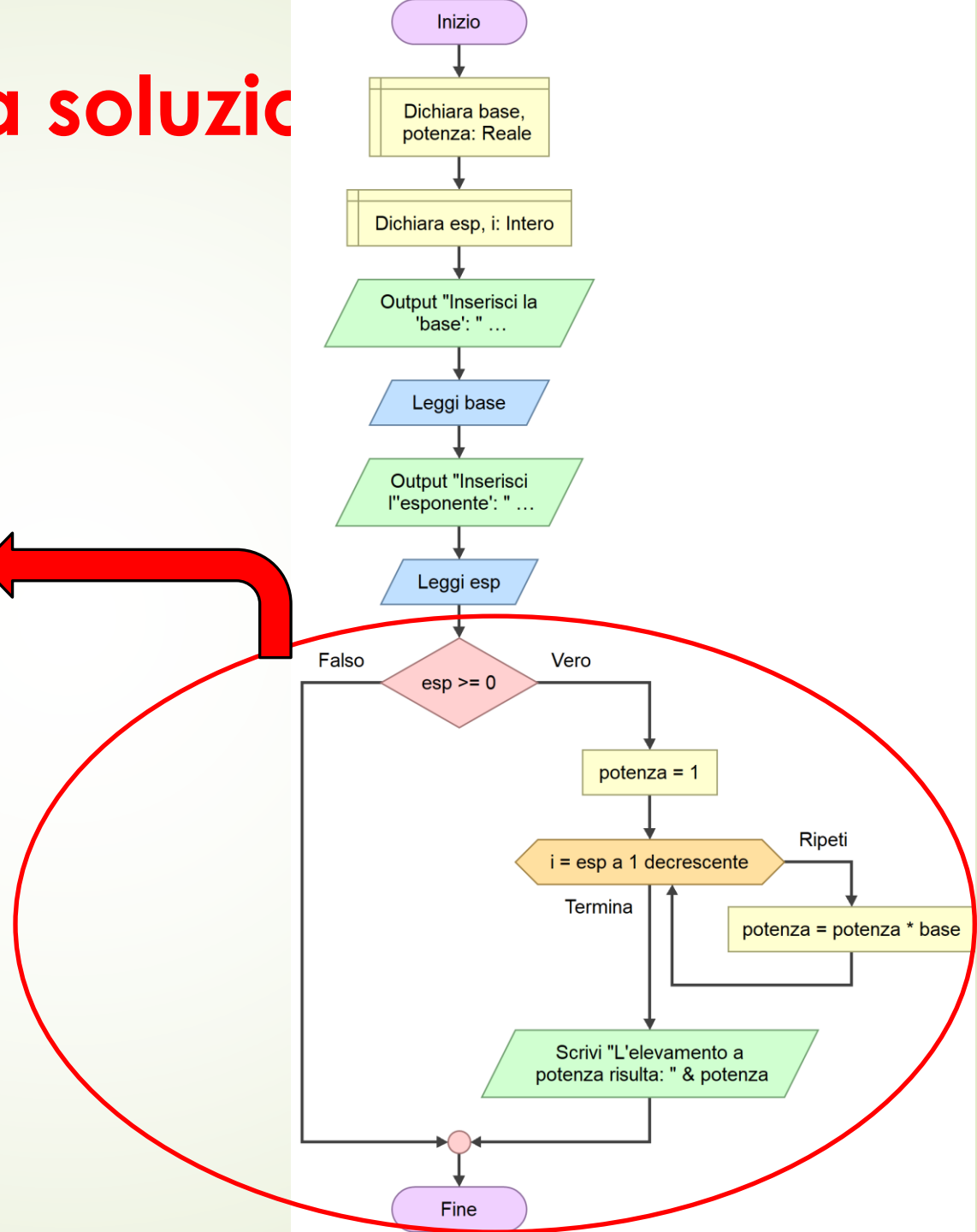
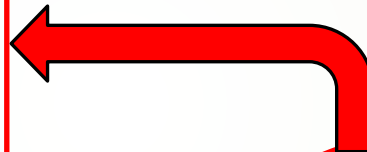
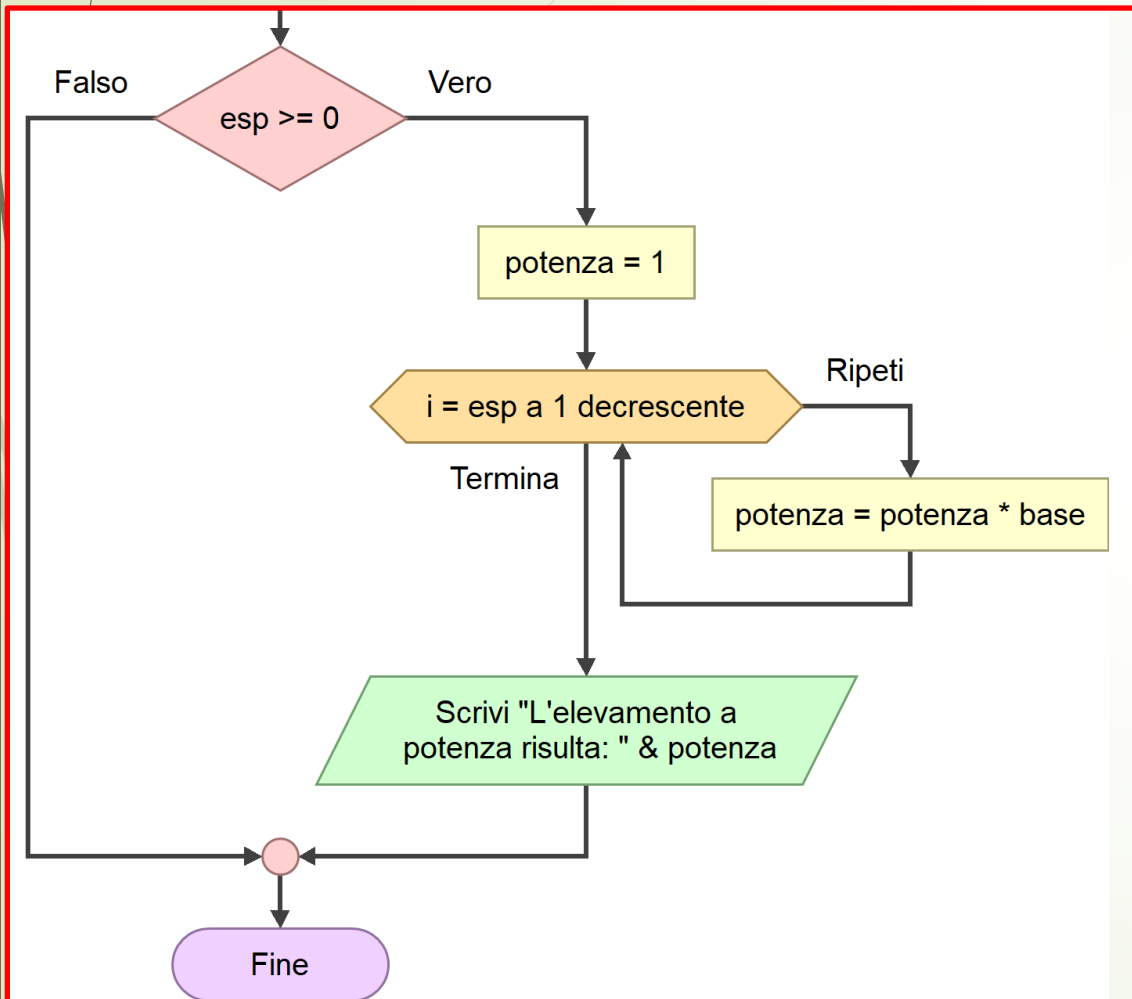
**=>**

**$3^2 = 9$**

## Idea (informale)

L' «esponente» diventa la variabile che conta le volte che bisogna moltiplicare la «base» per ottenere il risultato.

# PW: Progetto della soluzione



# PW: Realizzazione in Python

esponente.py > ...

```
1  print("Inserisci la 'base': ", end='', flush=True)
2  base = float(input())
3  print("Inserisci l''esponente': ", end='', flush=True)
4  esp = int(input())
5  if esp >= 0:
6      potenza = 1
7      for i in range(esp, 0, -1): # range: l'estremo inferiore '0' non è incluso
8          potenza = potenza * base
9      print("L'elevamento a potenza risulta: " + str(potenza))
```

# Project Work - Soluzione

```
Esponenziale.java
1 import java.io.*;
2
3 public class Esponenziale {
4
5     public static void main(String[] args) throws IOException{
6         double base = 0;
7         double potenza = 1;
8         int esp, i;
9         String str;
10
11         InputStreamReader input = new InputStreamReader(System.in);
12         BufferedReader tastiera = new BufferedReader(input);
13
14         System.out.print("Inserisci la 'base': ");
15         str = tastiera.readLine();
16         base = Double.parseDouble(str);
17
18         System.out.print("Inserisci l' 'esponente': ");
19         str = tastiera.readLine();
20         esp = Integer.parseInt(str);
21         System.out.print("");
22
23         if(esp >= 0) {
24             for(i = esp; i > 0; i--) {
25                 potenza = potenza * base;
26             }
27             System.out.print("L'elevamento a potenza risulta: " + potenza);
28         }
29     }
30 }
```



# **IDLE di Python (overview)**



# Configurazione Minimale per Python

Per **scrivere** un file sorgente in Python potete usare qualsiasi editor di testi (per esempio Notepad++).

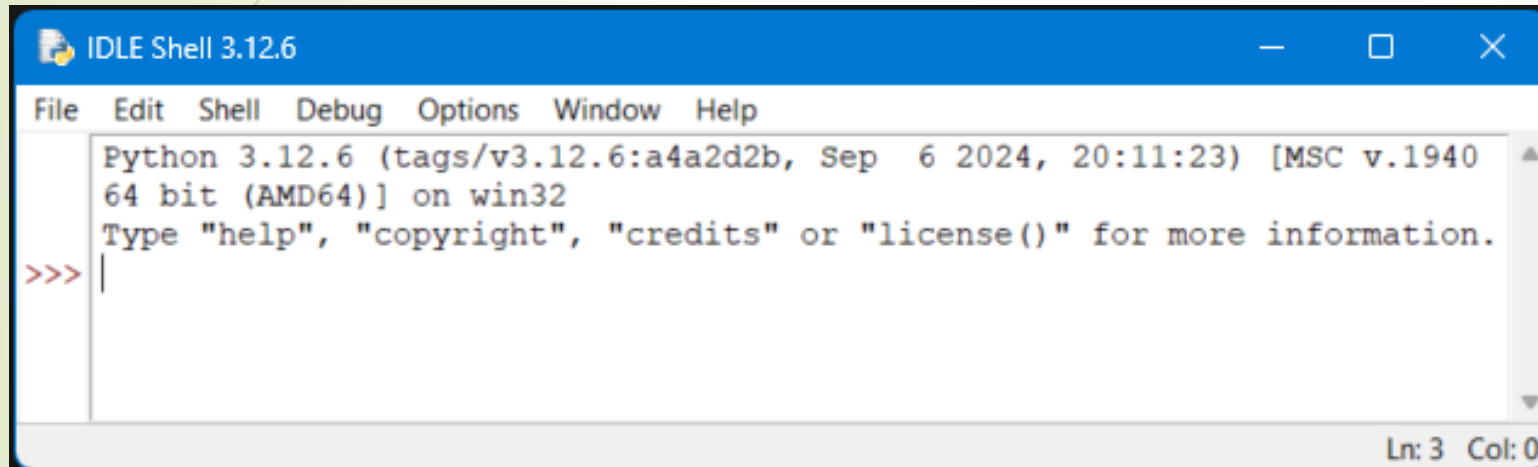
Mentre per **eseguire** file sorgenti («nomefile».py), il prompt della shell Python non è molto adatto, per cui l'installazione del linguaggio provvede di default un semplice ambiente integrato, chiamato **IDLE**: Integrated Development and Learning Environment.

Può essere eseguito da riga di comando in una shell:

```
"C:\...\Python3xx\pythonw.exe" "C:\...\Python3xx\Lib\idlelib\idle.pyw"
```

Oppure (metodo più semplice) cercando l'app installata ed eseguendola con un click...

# Python IDLE

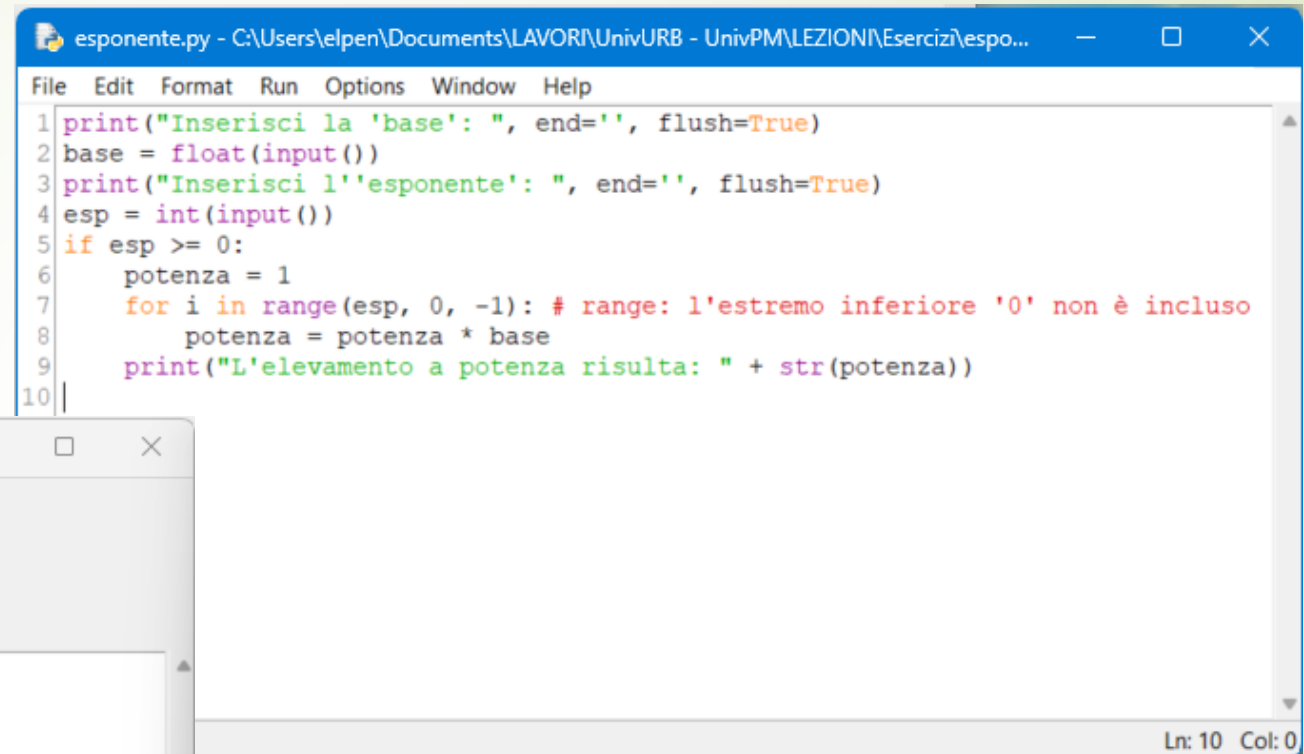


The screenshot shows the IDLE Shell 3.12.6 window. The title bar is blue with the text "IDLE Shell 3.12.6" and standard window controls. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main text area displays the Python 3.12.6 startup message: "Python 3.12.6 (tags/v3.12.6:a4a2d2b, Sep 6 2024, 20:11:23) [MSC v.1940 64 bit (AMD64)] on win32". Below this, it says "Type 'help', 'copyright', 'credits' or 'license()' for more information." followed by the prompt ">>>". The status bar at the bottom right shows "Ln: 3 Col: 0".

```
Python 3.12.6 (tags/v3.12.6:a4a2d2b, Sep 6 2024, 20:11:23) [MSC v.1940  
64 bit (AMD64)] on win32  
Type "help", "copyright", "credits" or "license()" for more information.  
>>> |
```



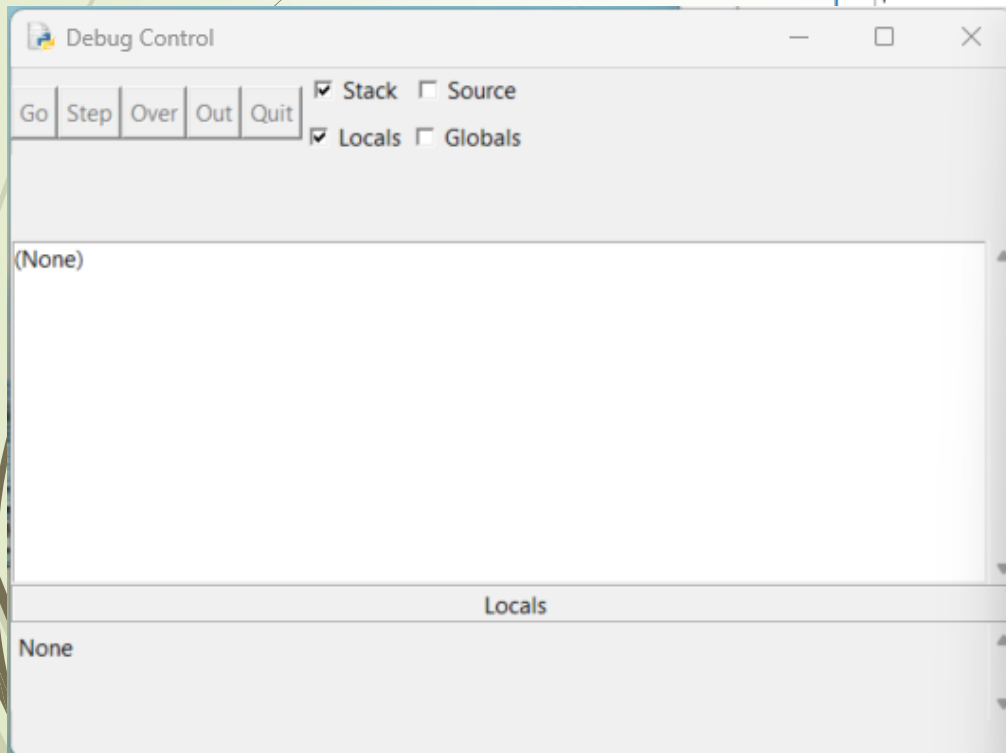
# IDLE: Editor e Debugger



The screenshot shows the IDLE Python editor window with the file 'esponente.py' open. The code is as follows:

```
1 print("Inserisci la 'base': ", end='', flush=True)
2 base = float(input())
3 print("Inserisci l''esponente': ", end='', flush=True)
4 esp = int(input())
5 if esp >= 0:
6     potenza = 1
7     for i in range(esp, 0, -1): # range: l'estremo inferiore '0' non è incluso
8         potenza = potenza * base
9     print("L'elevamento a potenza risulta: " + str(potenza))
10
```

The status bar at the bottom right indicates 'Ln: 10 Col: 0'.





**GRAZIE!**