CP020001 Computer Programming

Lecture: Flowchart and Pseudocode (Part I)

teerapong.pa@chula.ac.th

Reference:

1. https://colab.research.google.com/notebooks/
2. https://problemsolvingwithpython.com/
3. https://www.cp.eng.chula.ac.th/books/python101/
4. https://www.eng.chula.ac.th/th/20535
5. https://towardsdatascience.com/graph-visualisation-basics-with-python-part-i-flowcharts-6298c4f412e0
6. https://www.programiz.com/article/flowchart-programming
7. https://www.toppr.com/guides/python-guide/tutorials/python-flow-control/if-elif-else/python-if-if-else-if-elif-else-and-nested-if-statement/

About Me

Kao Panboonyuen

kao-panboonyuen

AI Research Scientist

Name: Teerapong Panboonyuen (P’Kao)

Contact: teerapong.pa@chula.ac.th

panboonyuen.kao@gmail.com

Education: Ph.D. (AI) Chula

Position: AI Team Lead, MARS

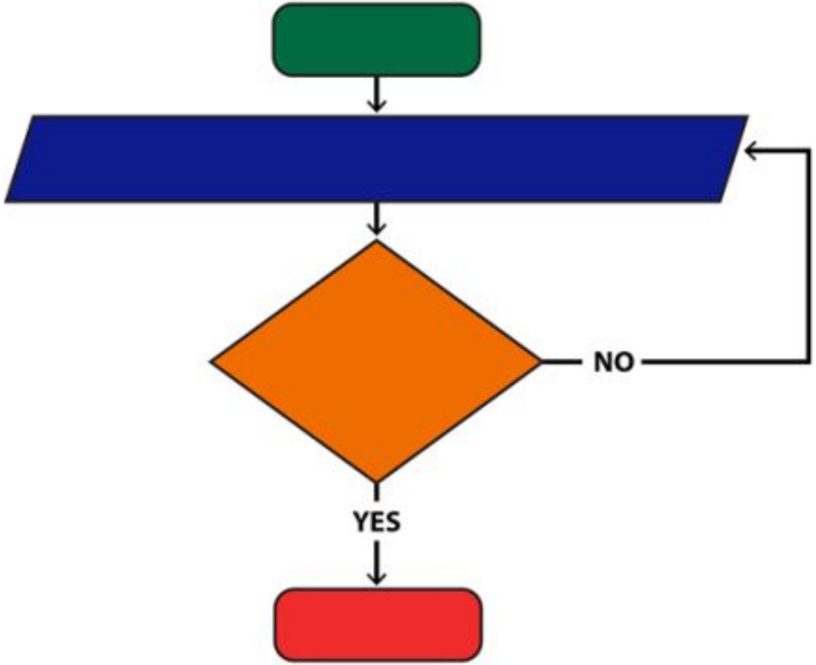
PostDoc, Chula

Interests: Computer Vision, Deep Learning

Machine Learning, Remote Sensing

Flowcharts and Programming

learning strategies and tips

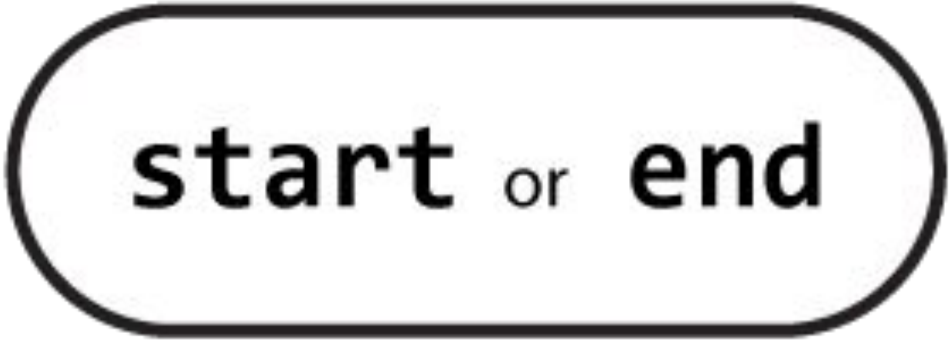


TECHNOKids

https://blog.technokids.com/programming/how-to-make-a-flowchart-for-programming-easy-to-understand/

Introduce Flowchart Shapes

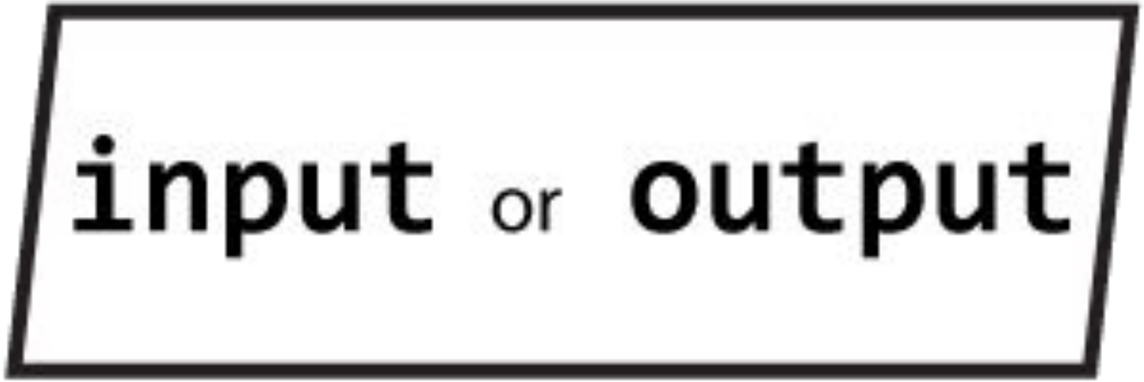
Use an oval to mark the beginning and end of the program.



https://blog.technokids.com/programming/how-to-make-a-flowchart-for-programming-easy-to-understand/

Introduce Flowchart Shapes

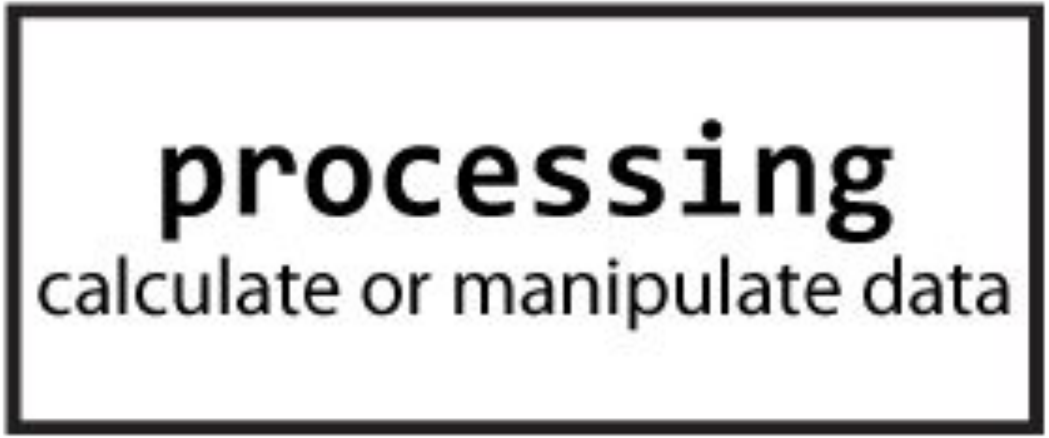
Use a parallelogram to show input or output. Input could be the user entering information. Output could be a message on the screen.



https://blog.technokids.com/programming/how-to-make-a-flowchart-for-programming-easy-to-understand/

Introduce Flowchart Shapes

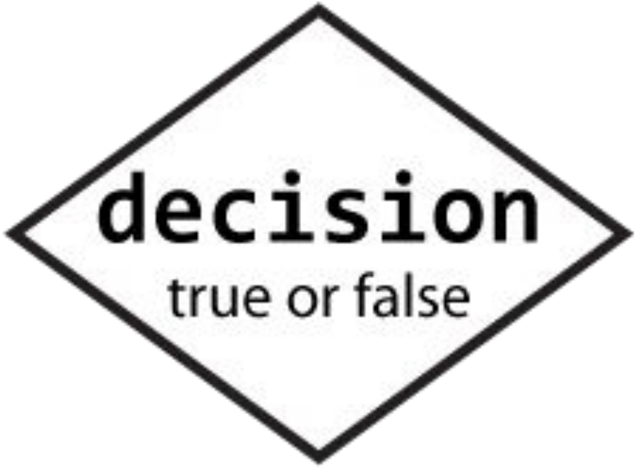
Use a rectangle to process an action. It could be a formula to calculate a value, or it might be a step that modifies text.



https://blog.technokids.com/programming/how-to-make-a-flowchart-for-programming-easy-to-understand/

Introduce Flowchart Shapes

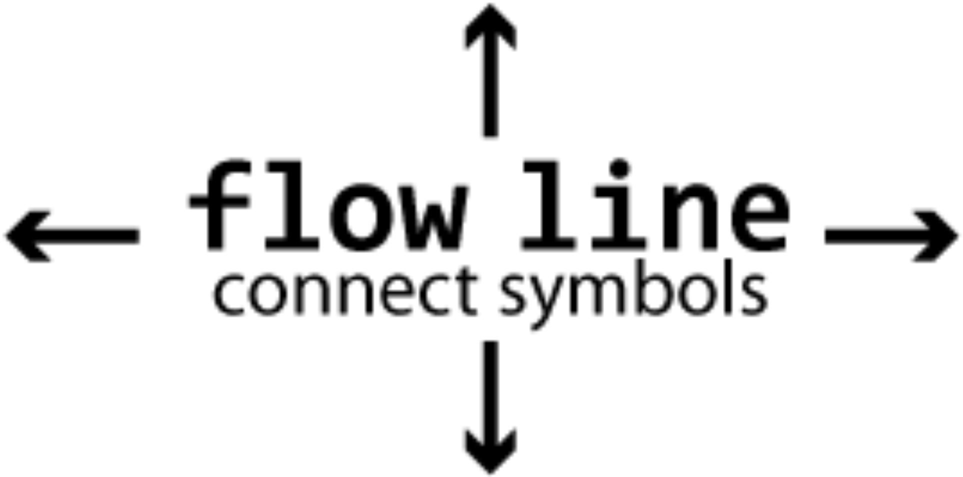
Use a diamond to make decisions. This shape will have two or more lines that come from it – one for each outcome. This step might ask a question or provide options. The result could be true or false, yes or no, or choices (red, blue, or green).



https://blog.technokids.com/programming/how-to-make-a-flowchart-for-programming-easy-to-understand/

Introduce Flowchart Shapes

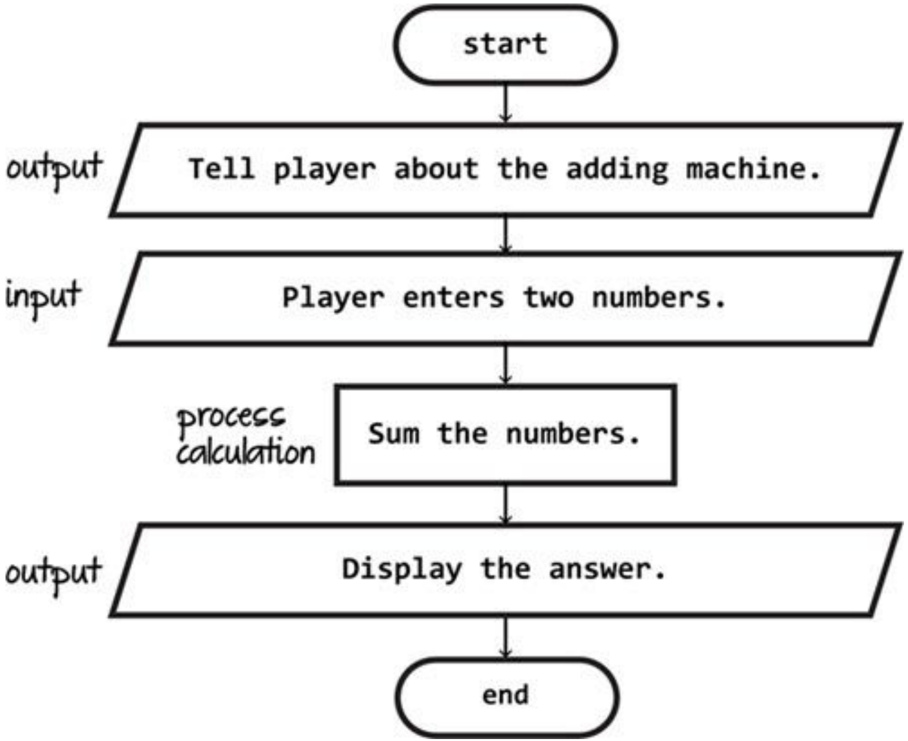
Use lines to connect the shapes. The arrows show the direction of the steps. Some lines should include labels, such as yes or no, to explain what is happening.



https://blog.technokids.com/programming/how-to-make-a-flowchart-for-programming-easy-to-understand/

Start Simple with a Familiar Task

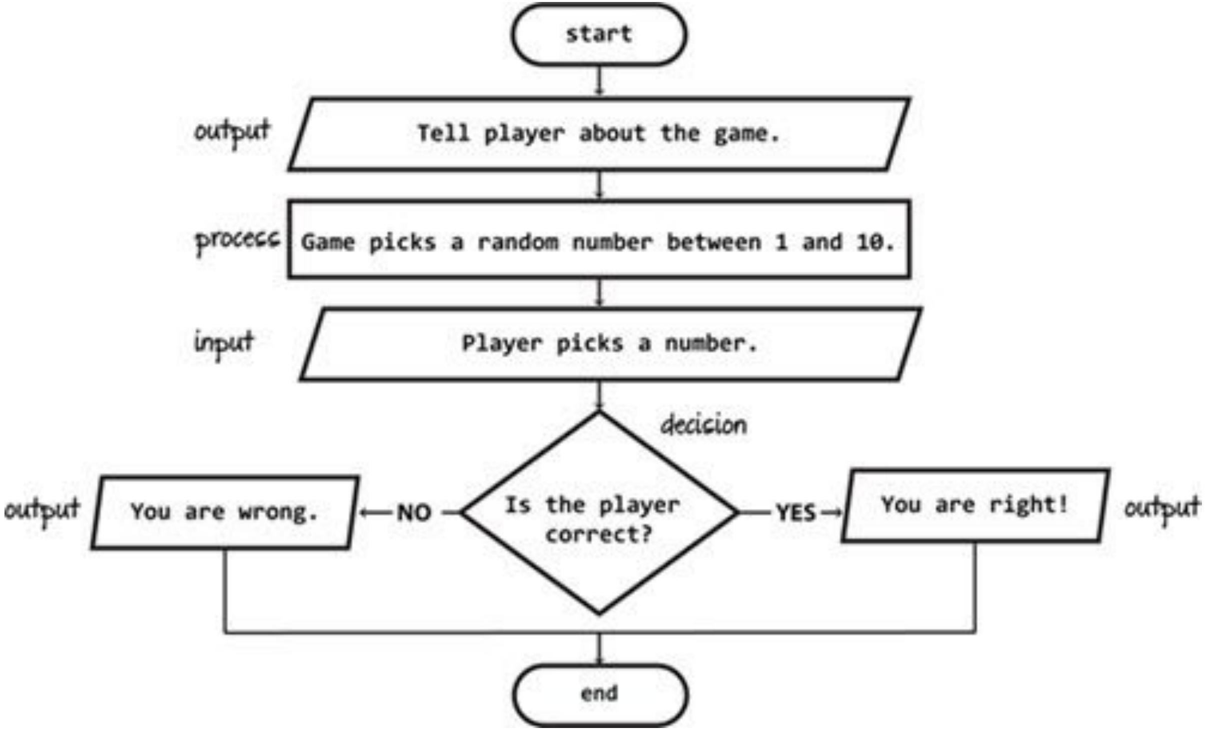
A calculator is something familiar. It is on every phone and computer. Since it is recognizable, student can transfer their existing knowledge to the task. This enables them to focus their attention towards organizing the program components into a flowchart.



https://blog.technokids.com/programming/how-to-make-a-flowchart-for-programming-easy-to-understand/

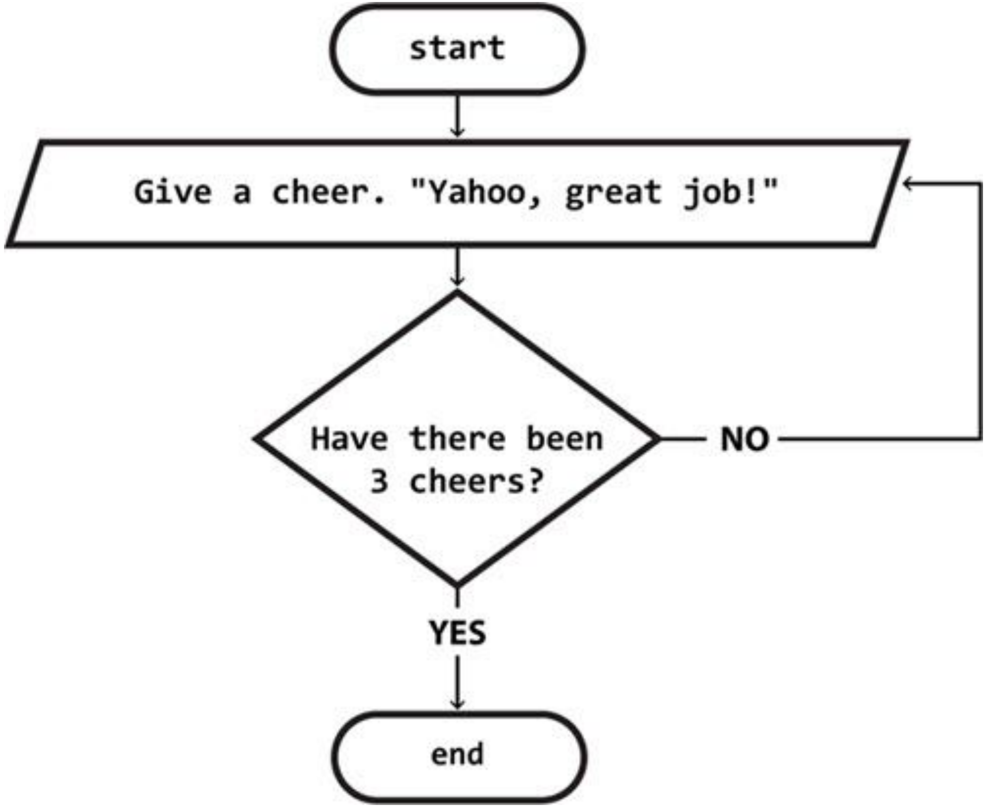
Decision Making Sample: A guessing game.

Players must correctly pick a number between 1 and 10 to win. Although this program only has a few steps, it uses all the standard shapes in a flowchart and has branches.



Apply Pattern Recognition to Loop Instructions

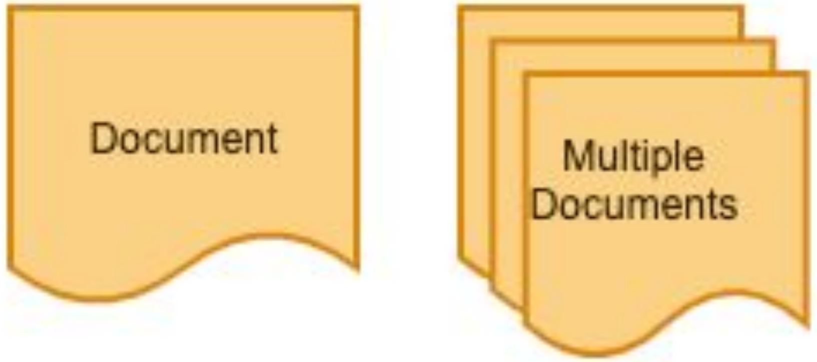
A simple cheer is a great place to start making loops in a flowchart. Now that students understand the meaning of the standard shapes and branches, they can focus upon the direction of the lines. Since only the output is repeated, there is a simplicity to the task that is ideal for beginners.



Intermediate & Advanced Flowchart Symbols

Document Symbols

Single and multiple document icons show that there are additional points of reference involved in your flowchart. You might use these to indicate items like “create an invoice” or “review testing paperwork.”

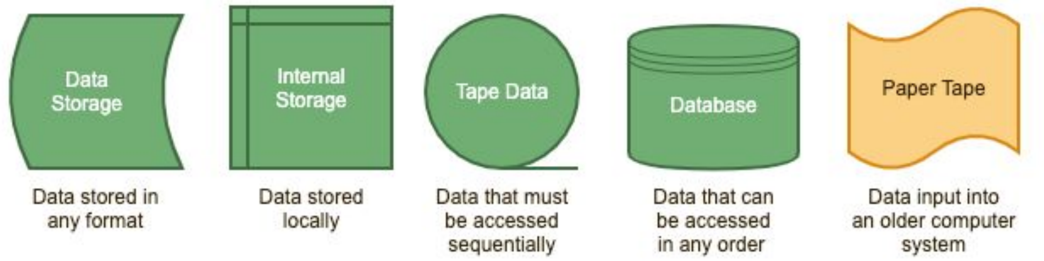


https://www.gliffy.com/blog/guide-to-flowchart-symbols

Intermediate & Advanced Flowchart Symbols

Data Symbols

Data symbols clarify where the data your flowchart references is being stored. (You probably won't use the paper tape symbol, but it definitely came in handy back in the day.)

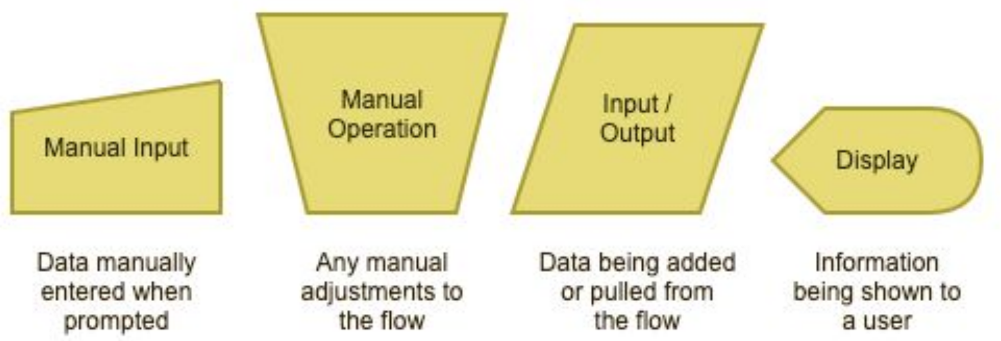

Data Storage
Data stored in any format
Internal Storage
Data stored locally
Tape Data
Data that must be accessed sequentially
Database
Data that can be accessed in any order
Paper Tape
Data input into an older computer system

https://www.gliffy.com/blog/guide-to-flowchart-symbols

Intermediate & Advanced Flowchart Symbols

Input & Output Symbols

Input and output symbols show where and how data is coming in and out throughout your process.

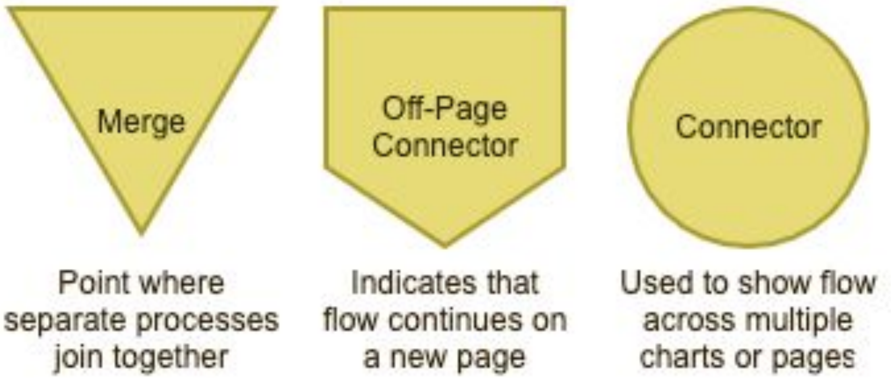

Manual Input
Data manually entered when prompted
Manual Operation
Any manual adjustments to the flow
Input / Output
Data being added or pulled from the flow
Display
Information being shown to a user

https://www.gliffy.com/blog/guide-to-flowchart-symbols

Intermediate & Advanced Flowchart Symbols

Merging & Connecting Symbols

Agreed-upon merging and connector symbols make it easier to connect flowcharts that span multiple pages.



Point where separate processes join together

Indicates that flow continues on a new page

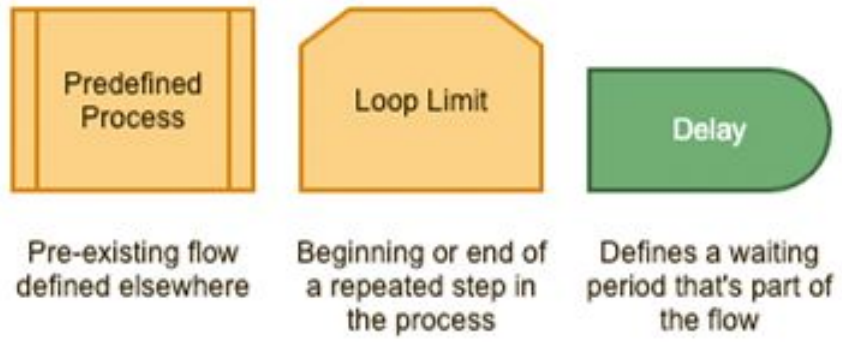
Used to show flow across multiple charts or pages

https://www.gliffy.com/blog/guide-to-flowchart-symbols

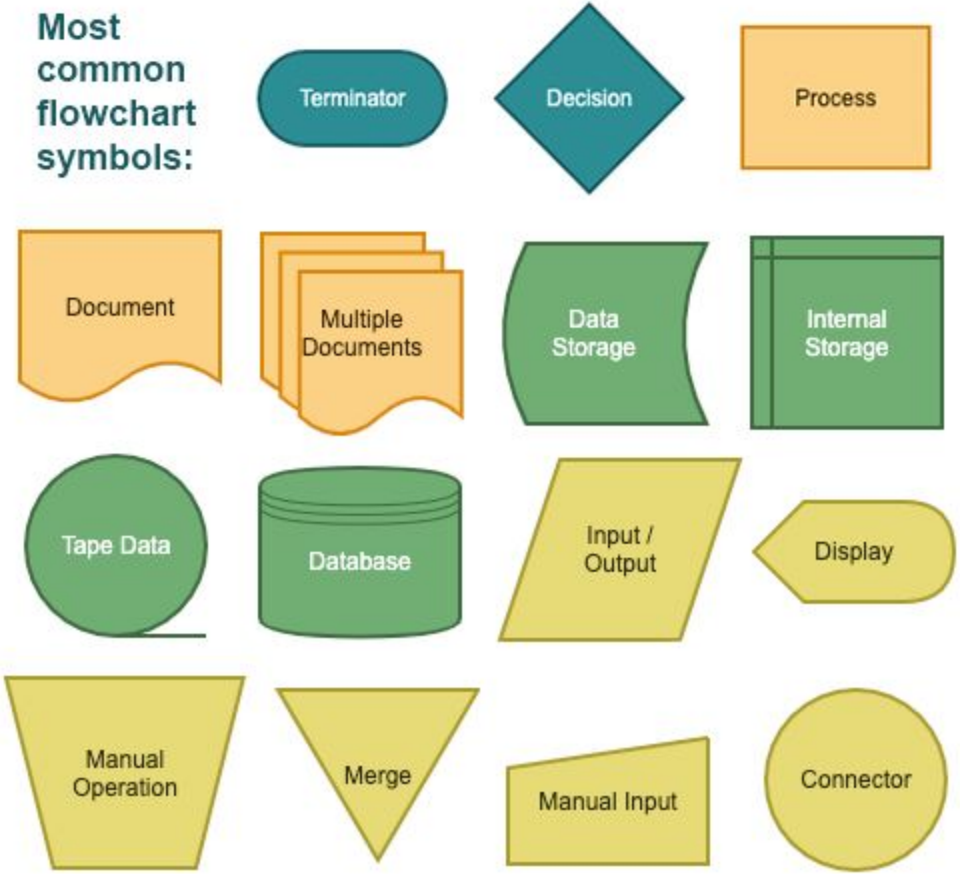
Intermediate & Advanced Flowchart Symbols

Additional Useful Flowchart Symbols

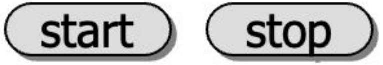
The above are a few additional symbols that prove your flowcharting prowess when put to good use.


Predefined Process
Loop Limit
Delay
Pre-existing flow defined elsewhere
Beginning or end of a repeated step in the process
Defines a waiting period that's part of the flow

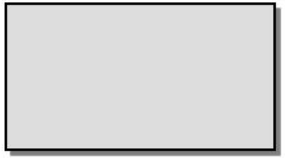
https://www.gliffy.com/blog/guide-to-flowchart-symbols



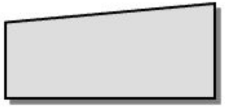
https://www.gliffy.com/blog/guide-to-flowchart-symbols



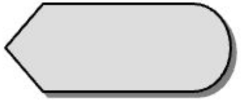
เริ่มต้น/สิ้นสุด



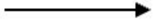
process



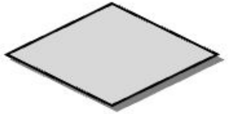
รับข้อมูลจากแป้นพิมพ์



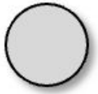
แสดงผลออกทาง
จอภาพ



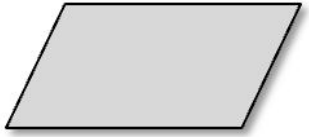
เส้นทาง



decision

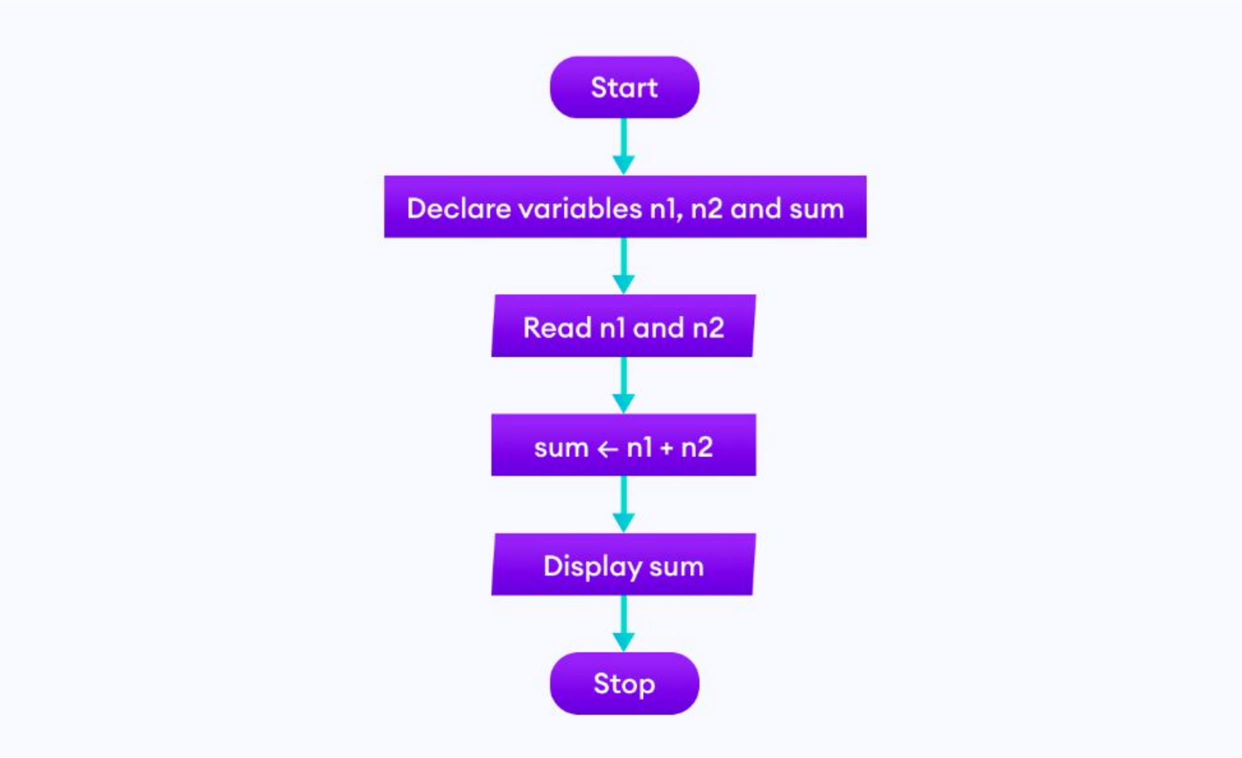


จุดเชื่อม

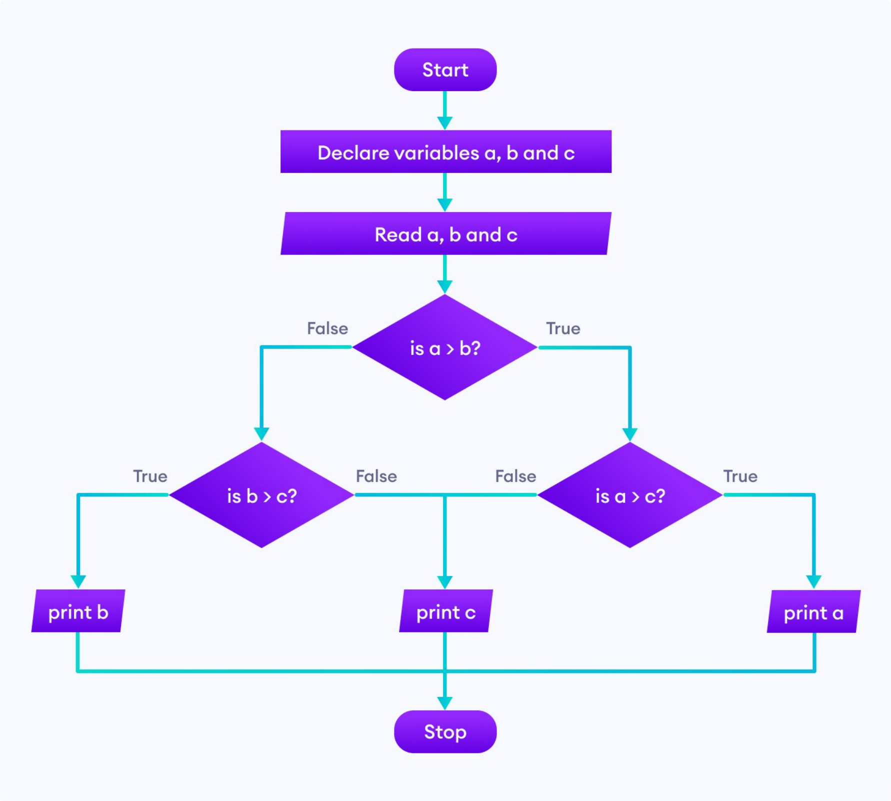


input/output

Examples of flowcharts in programming



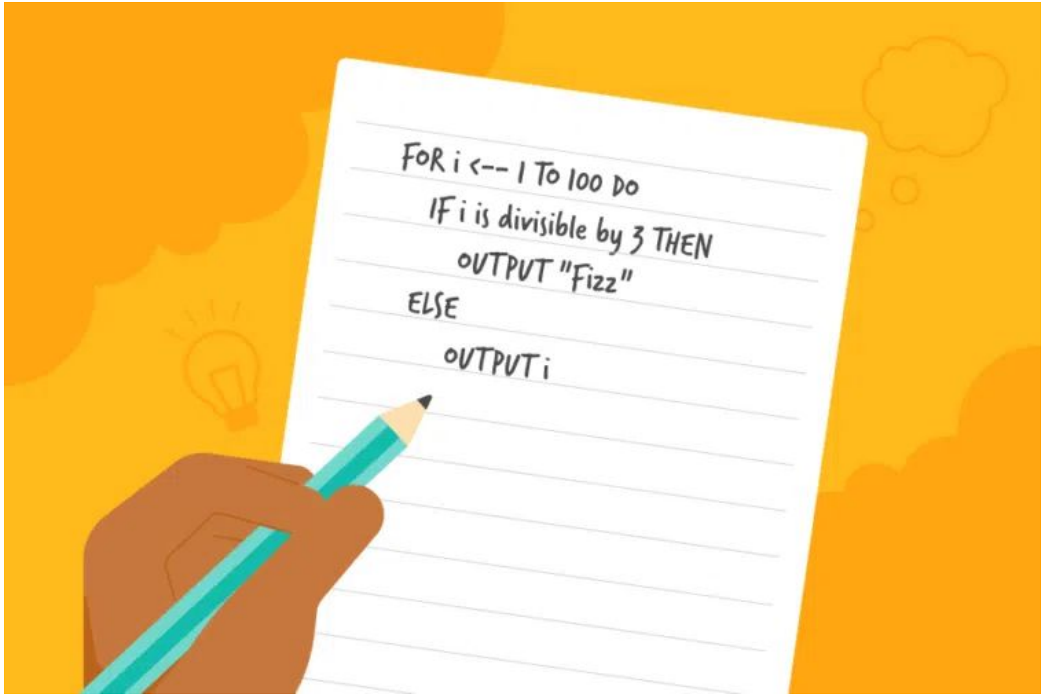
Find the largest among three different numbers entered by the user.



Pseudocode

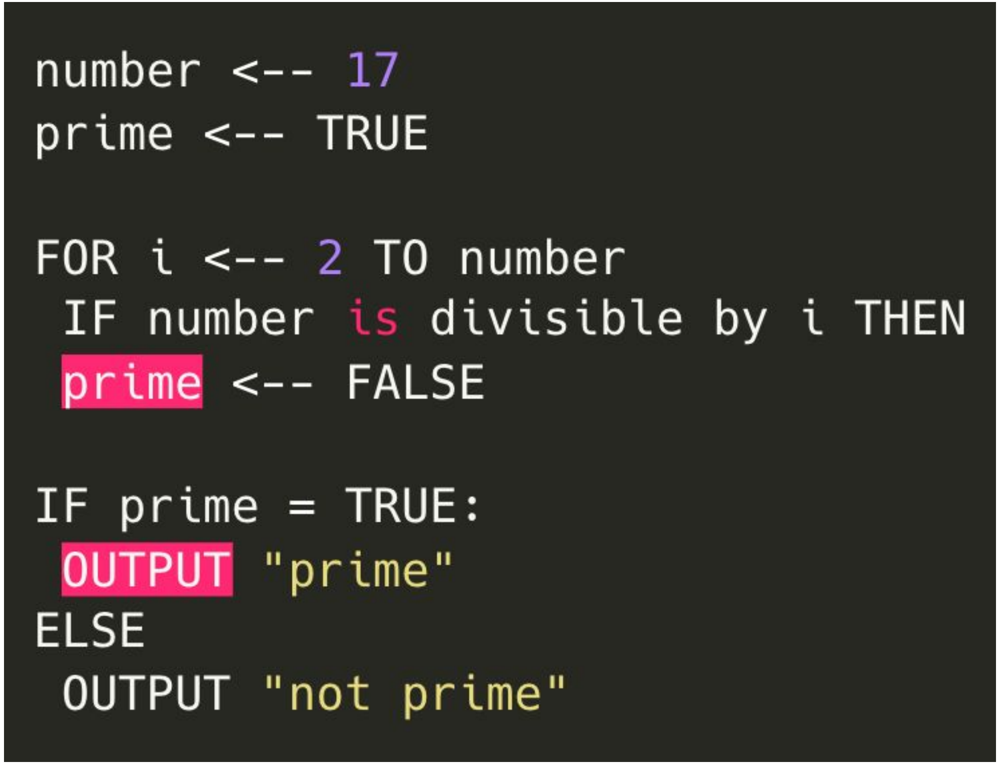
What is Pseudocode?

- a way of expressing an algorithm **without** conforming to specific syntax rules.
- By learning to read and write pseudocode, you can easily communicate ideas and concepts to other programmers, even though they may be using completely different languages.



```
FOR i <-- 1 TO 100 DO
  IF i is divisible by 3 THEN
    OUTPUT "Fizz"
 ELSE
    OUTPUT i
```

Sample: A prime number

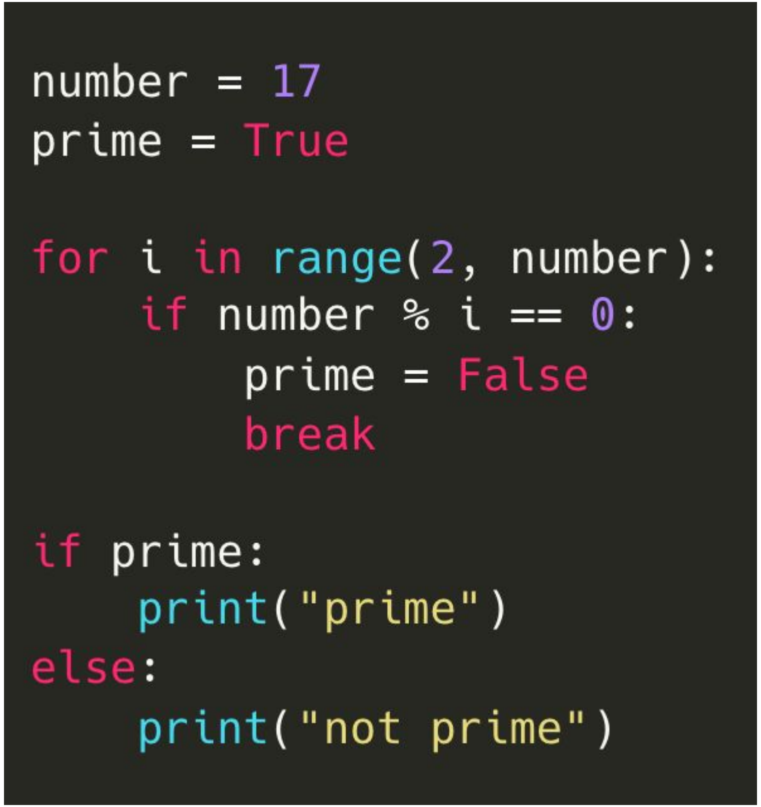


```
number <-- 17
prime <-- TRUE

FOR i <-- 2 TO number
 IF number is divisible by i THEN
 prime <-- FALSE

IF prime = TRUE:
 OUTPUT "prime"
ELSE
 OUTPUT "not prime"
```

Pseudo Code

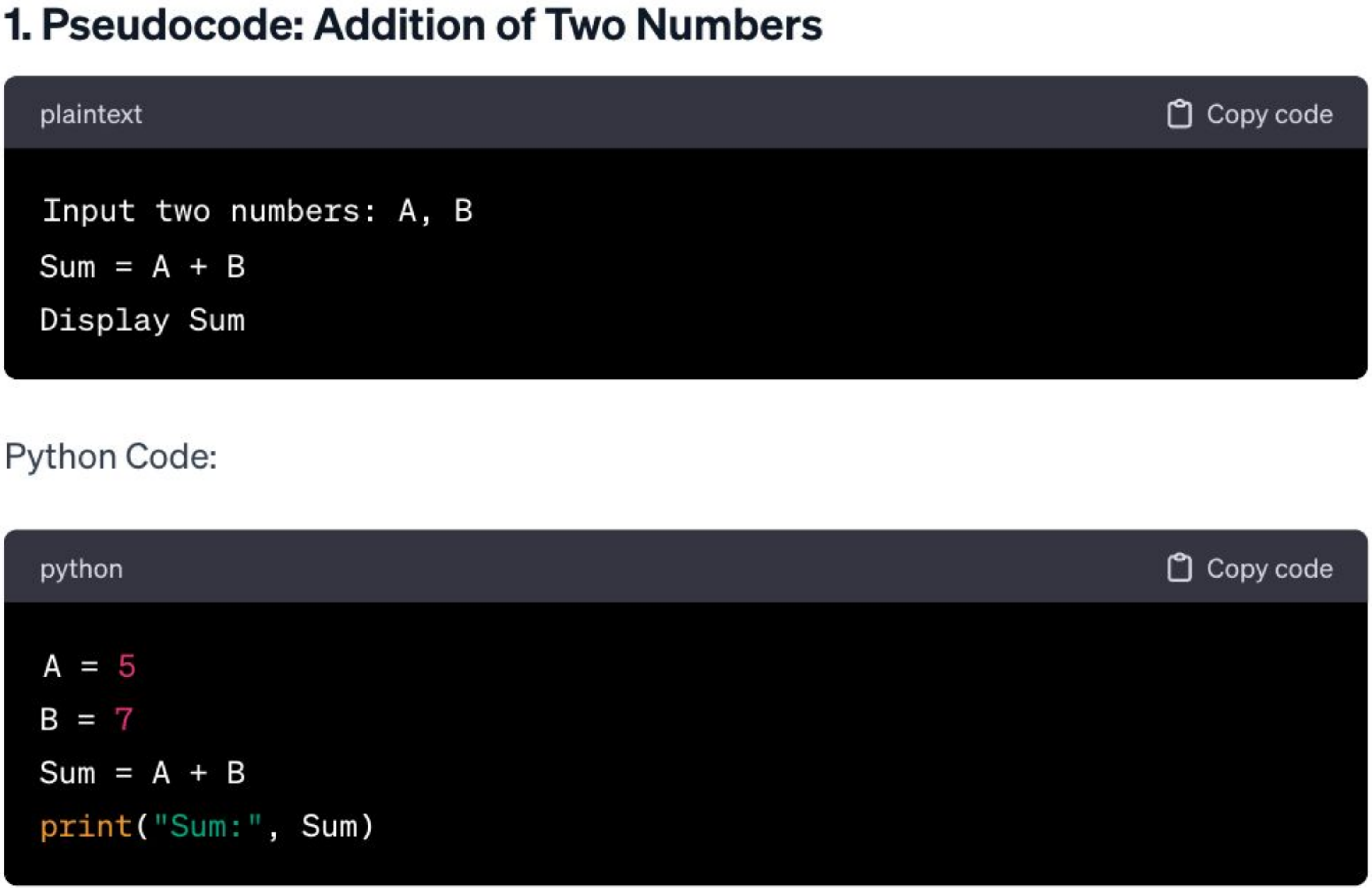


```python
number = 17
prime = True

for i in range(2, number):
    if number % i == 0:
        prime = False
        break

if prime:
    print("prime")
else:
    print("not prime")
```

Python Code

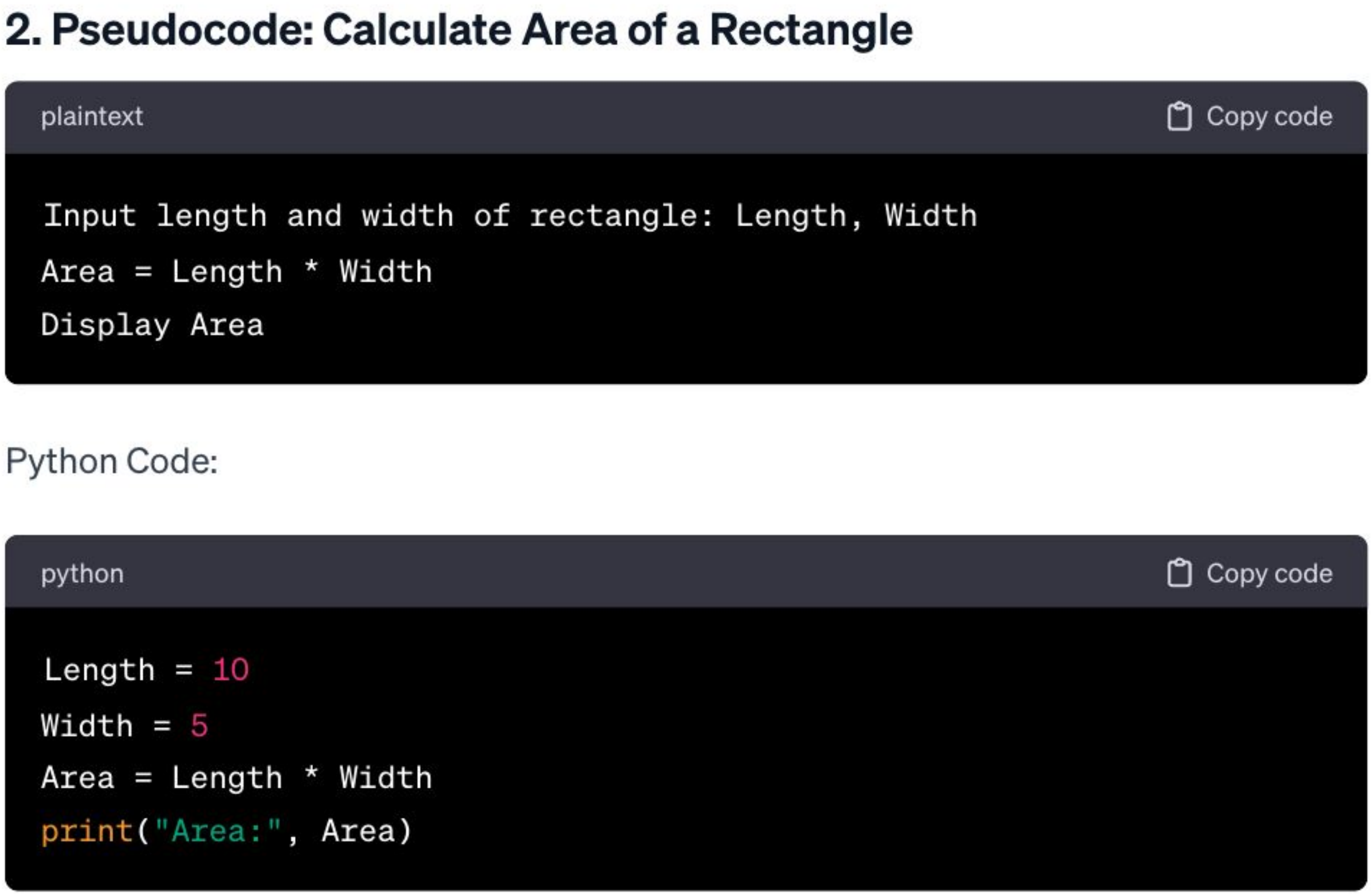


1. Pseudocode: Addition of Two Numbers

```
Input two numbers: A, B
Sum = A + B
Display Sum
```

Python Code:

```python
A = 5
B = 7
Sum = A + B
print("Sum:", Sum)
```

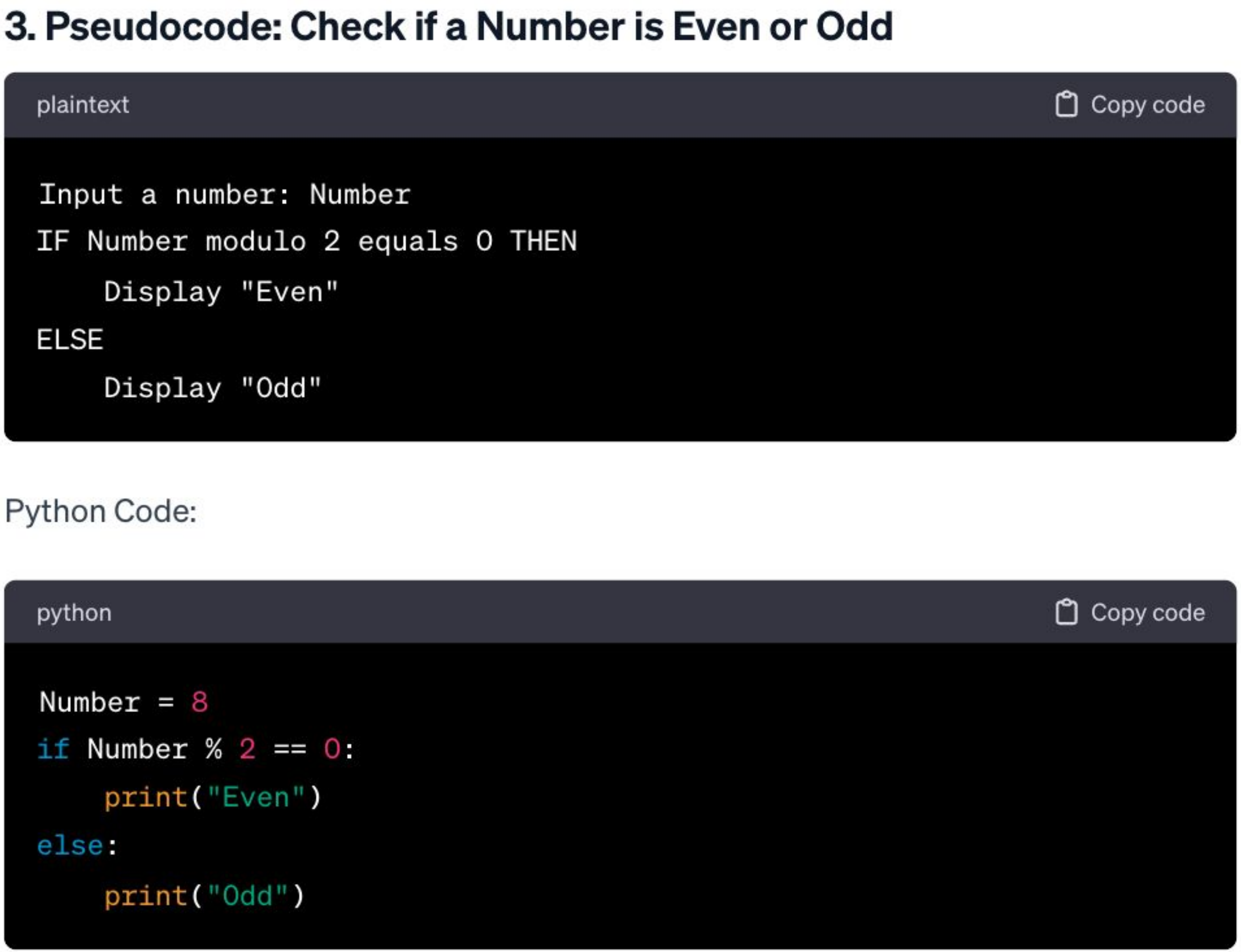


2. Pseudocode: Calculate Area of a Rectangle

```
Input length and width of rectangle: Length, Width
Area = Length * Width
Display Area
```

Python Code:

```python
Length = 10
Width = 5
Area = Length * Width
print("Area:", Area)
```

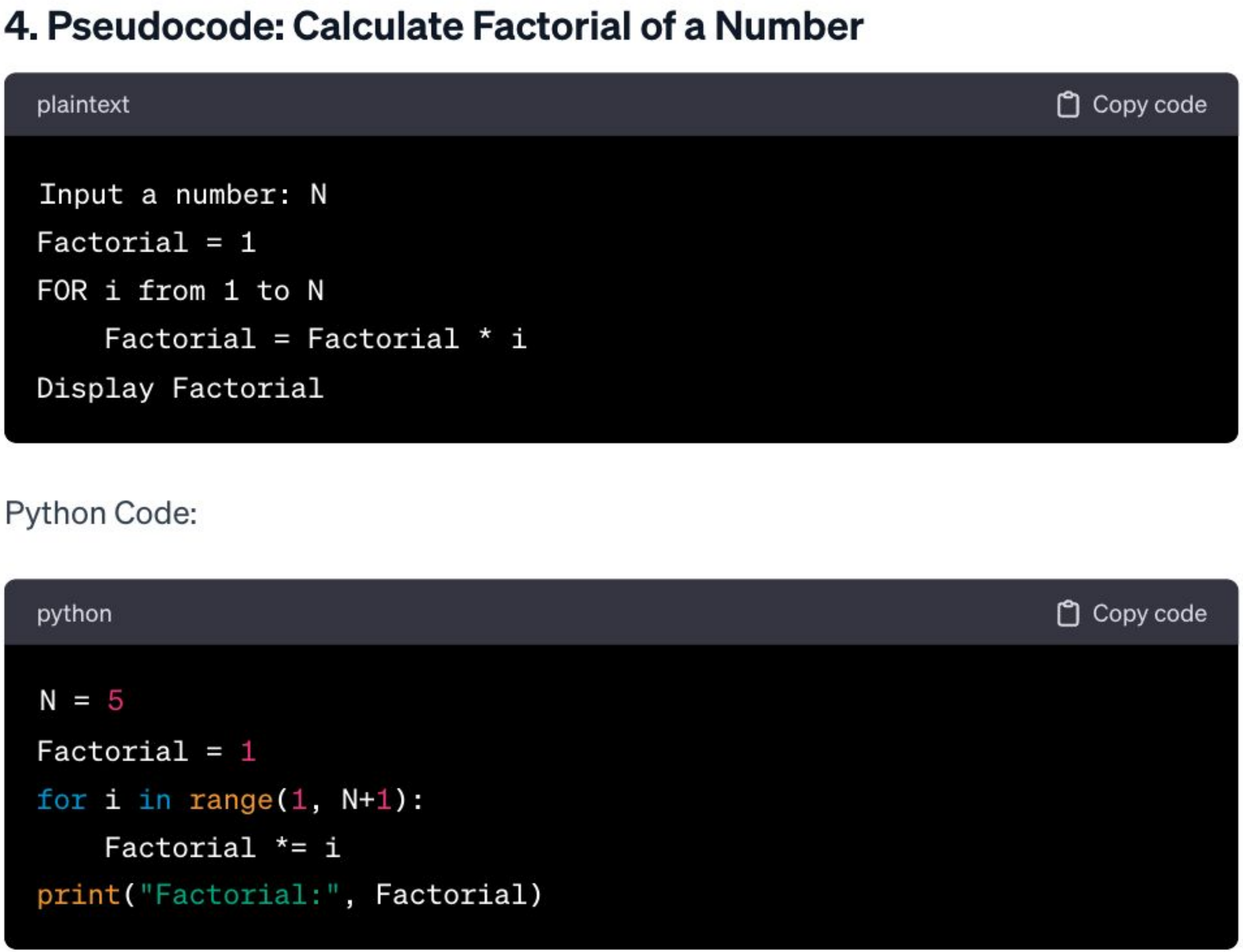


3. Pseudocode: Check if a Number is Even or Odd

```
Input a number: Number
IF Number modulo 2 equals 0 THEN
    Display "Even"
ELSE
    Display "Odd"
```

Python Code:

```python
Number = 8
if Number % 2 == 0:
    print("Even")
else:
    print("Odd")
```



4. Pseudocode: Calculate Factorial of a Number

```plaintext
Input a number: N
Factorial = 1
FOR i from 1 to N
    Factorial = Factorial * i
Display Factorial
```

Python Code:

```python
N = 5
Factorial = 1
for i in range(1, N+1):
    Factorial *= i
print("Factorial:", Factorial)
```