

Name: **HIDAYAT ULLAH**

R No: **B24F0050AI099**

Assignment: **01**

Date: **20/09/2025**

Submitted to: **Mr, Umm-e-Habiba**

Task 1: Using if with := (Walrus Operator)

Scenario.

Imagine you are designing a simple live polling system. The program should allow multiple users to vote either "yes" or "no", and it should stop only when a user types "stop".

Requirements:

- 1. Use the Walrus Operator (:=) to both read input and check it inside the while loop condition.
- 2. If the user enters "yes", increase the yes votes count.
- 3. If the user enters "no", increase the no votes count.
- 4. If the user enters anything else (other than "yes", "no", or "stop") → print "Invalid vote, try again."
- 5. At the end of polling, display the total number of "yes" and "no" votes.

```
yes_votes = 0
no_votes = 0

while (vote := input("Enter vote (yes/no or stop to finish): ").lower()) != "stop":
    if vote == "yes":
        yes_votes += 1
    elif vote == "no":
        no_votes += 1
    else:
        print("Invalid vote, try again.")
print("Voting Results:")
print("Yes votes =", yes_votes)
print("No votes =", no_votes)
```

```
Enter vote (yes/no or stop to finish): yes
Enter vote (yes/no or stop to finish): no
Enter vote (yes/no or stop to finish): maybe
Invalid vote, try again.
Enter vote (yes/no or stop to finish): yes
Enter vote (yes/no or stop to finish): stop
Voting Results:
Yes votes = 2
No votes = 1
```

Descriptions

This task make a voting system.
User can type yes, no or stop.
Program count yes and no votes, and show invalid if wrong word.
I use walrus operator (:=) inside while loop to take input and check same time.

✓

Task 2: Using Conditional Expressions inside Loops

Scenario.
Consider developing a shopping cart system. The prices of items are stored in a list. For each item:

1. If the price is ≥ 1000 , tag it as "Expensive".
2. Otherwise, tag it as "Affordable".
3. Use a for loop with a conditional expression (ternary operator) inside the loop.

```
cart_prices = [500, 1500, 200, 2500]

print("Cart Prices:", cart_prices)
for price in cart_prices:
    tag = "Expensive" if price >= 1000 else "Affordable"
    print("Item with price", price, "->", tag)
```

Cart Prices: [500, 1500, 200, 2500]
Item with price 500 -> Affordable
Item with price 1500 -> Expensive
Item with price 200 -> Affordable
Item with price 2500 -> Expensive

✓

Descriptions

This task make a shopping cart.
Prices are stored in a list.
If price ≥ 1000 program say Expensive, else Affordable.
I use for loop with short if else (conditional expression).



✓

Task 3: Using for loops with else clause

Scenario.
Suppose, you are building a cybersecurity file scanner. The system checks a list of filenames for any suspicious ones (e.g., "virus.exe", "malware.bat", "trojan.py"). Requirements:

1. Store suspicious keywords in a list.
2. Use a for loop with an else clause to scan filenames one by one.
3. If a suspicious file is found, print "Warning: is malicious" and stop the scan.
4. If the loop finishes without finding any suspicious files, the else clause should print "System Clean: No threats detected."

```
files = ["assignment.py", "slides.pptx", "notes.txt"]
suspicious = ["virus.exe", "malware.bat", "trojan.py"]

for file in files:
    if file in suspicious:
        print("Warning:", file, "is malicious")
        break
else:
    print("System Clean: No threats detected.")
```

System Clean: No threats detected.

▼

Descriptions

This task is file scanner.
It check every file from list one by one.
If suspicious file found, program show warning and stop.
If not found, program say system clean using for loop with else.



▼

Task 4: Using any() and all() with Conditions

Scenario.
Let us suppose, you are testing an AI model that outputs confidence scores (between 0 and 1) for its predictions. Your task is to check the quality of these predictions. Requirements:

1. Store model confidence scores in a list (e.g., [0.92, 0.85, 0.78, 0.60]).
2. If all confidence scores $\geq 0.80 \rightarrow$ print "Model is reliable for deployment."
3. If any confidence score $< 0.50 \rightarrow$ print "Warning: Model predictions are too weak."
4. Otherwise, print "Model requires further tuning."

```
confidence_scores = [0.92, 0.85, 0.81, 0.88]
print("Confidence Scores:", confidence_scores)

if all(score >= 0.80 for score in confidence_scores):
    print("Model is reliable for deployment.")
elif any(score < 0.50 for score in confidence_scores):
    print("Warning: Model predictions are too weak.")
else:
    print("Model requires further tuning.")
```

Confidence Scores: [0.92, 0.85, 0.81, 0.88]
Model is reliable for deployment.

▼

Descriptions

This task check AI model confidence scores.
Scores are between 0 and 1.
If all scores $\geq 0.80 \rightarrow$ model reliable, if any score $< 0.50 \rightarrow$ model weak.
Otherwise program say model need tuning using any() and all().



Summary of All Tasks

In this assignment I finish 4 tasks using Python control flow.

Task 1 was voting system using walrus operator to take input and count yes/no.

Task 2 was shopping cart using conditional expression to check price expensive or affordable.

Task 3 was file scanner using for-else to find suspicious files and give warning or system clean.

Task 4 was model confidence checker using any() and all() to test scores and give result.

These tasks help me learn special Python features which make code short and clear.