# ASSIGNMENT - 6.3

KONDAPALLI DEVI PRIYANKA
2303A51108

TASK - 1:
Loops – Automorphic Numbers in a Range)
• Task: Prompt AI to generate a function that displays all Automorphic numbers between 1 and 1000 using a for loop.
• Instructions:
o Get AI-generated code to list Automorphic numbers using a for loop.
o Analyze the correctness and efficiency of the generated logic.
o Ask AI to regenerate using a while loop and compare both implementations.
Expected Output #1:
• Correct implementation that lists Automorphic numbers using both loop types, with explanation

PROMPT:

```
#display all the automorphic number in the range of 1 to 1000 bby using
FOR loop and WHILE loop
# Using FOR loop
```

CODE:

```
print("Automorphic numbers between 1 and 1000 using FOR loop:")
for num in range(1, 1001):
    square = num ** 2
    if str(square).endswith(str(num)):
        print(num)

# Using WHILE loop
print("\nAutomorphic numbers between 1 and 1000 using WHILE loop:")
num = 1
while num <= 1000:
    square = num ** 2
    if str(square).endswith(str(num)):
        print(num)
```

```
    num += 1

#calculate the total time taken by each loop
import time
# Time taken by FOR loop
start_time_for = time.time()
for num in range(1, 1001):
    square = num ** 2
    if str(square).endswith(str(num)):
        pass
end_time_for = time.time()
for_loop_time = end_time_for - start_time_for
print(f"\nTime taken by FOR loop: {for_loop_time} seconds")
# Time taken by WHILE loop
start_time_while = time.time()
num = 1
while num <= 1000:
    square = num ** 2
    if str(square).endswith(str(num)):
        pass
    num += 1
end_time_while = time.time()
while_loop_time = end_time_while - start_time_while
print(f"Time taken by WHILE loop: {while_loop_time} seconds")

#print the conclusion about the time taken by each loop with the reason
if for_loop_time < while_loop_time:
    print("FOR loop is faster than WHILE loop.")
elif while_loop_time < for_loop_time:
    print("WHILE loop is faster than FOR loop.")
else:
    print("Both loops took the same amount of time.")
```

EXPLANATION:
Same logic, different control flow.
`for` is cleaner for fixed ranges; `while` gives manual control.
Truth: **for-loop is better here**.

**Task 2: Online Shopping Feedback Classification**

```
# by using the if-elif-else display the feedback of the online shopping
based numerical rating (1-5)
# 1 - Very Poor, 2 - Poor, 3 - Average, 4 - Good, 5 - Excellent
# get the rating from the user
# Using if-elif-else
# get the rating from the user
rating = int(input("Enter your rating for the online shopping (1-5): "))
if rating == 1:
    print("Feedback: Very Poor")
elif rating == 2:
    print("Feedback: Poor")
elif rating == 3:
    print("Feedback: Average")
elif rating == 4:
    print("Feedback: Good")
elif rating == 5:
    print("Feedback: Excellent")
else:
    print("Invalid rating! Please enter a number between 1 and 5.")
# Using dictionary mapping
# get the rating from the user
rating = int(input("Enter your rating for the online shopping (1-5): "))
feedback_dict = {
    1: "Very Poor",
    2: "Poor",
    3: "Average",
    4: "Good",
    5: "Excellent"
}
feedback = feedback_dict.get(rating, "Invalid rating! Please enter a
number between 1 and 5.")
print(f"Feedback: {feedback}")
# Using match-case (Python 3.10+)
# get the rating from the user
rating = int(input("Enter your rating for the online shopping (1-5): "))
match rating:
    case 1:
        print("Feedback: Very Poor")
    case 2:
```

```
        print("Feedback: Poor")
    case 3:
        print("Feedback: Average")
    case 4:
        print("Feedback: Good")
    case 5:
        print("Feedback: Excellent")
    case _:
        print("Invalid rating! Please enter a number between 1 and 5.")

#calculate the total time taken by each method
import time
# Time taken by if-elif-else
start_time_if = time.time()
rating = 3  # Example rating
if rating == 1:
    pass
elif rating == 2:
    pass
elif rating == 3:
    pass
elif rating == 4:
    pass
elif rating == 5:
    pass
else:
    pass
end_time_if = time.time()
if_time = end_time_if - start_time_if
print(f"\nTime taken by if-elif-else: {if_time} seconds")
# Time taken by dictionary mapping
start_time_dict = time.time()
rating = 3  # Example rating
feedback_dict = {
    1: "Very Poor",
    2: "Poor",
    3: "Average",
    4: "Good",
    5: "Excellent"
}
```

```
feedback = feedback_dict.get(rating, "Invalid rating! Please enter a
number between 1 and 5.")
end_time_dict = time.time()
dict_time = end_time_dict - start_time_dict
print(f"Time taken by dictionary mapping: {dict_time} seconds")
# Time taken by match-case
start_time_match = time.time()
rating = 3  # Example rating
match rating:
    case 1:
        pass
    case 2:
        pass
    case 3:
        pass
    case 4:
        pass
    case 5:
        pass
    case _:
        pass
end_time_match = time.time()
match_time = end_time_match - start_time_match
print(f"Time taken by match-case: {match_time} seconds")
#print the conclusion about the time taken by each method with the reason
min_time = min(if_time, dict_time, match_time)
if min_time == if_time:
    fastest = "if-elif-else"
elif min_time == dict_time:
    fastest = "dictionary mapping"
else:
    fastest = "match-case"
print(f"\nThe fastest method is {fastest}.")

#give the justification for the fastest method
if fastest == "dictionary mapping":
    print("Dictionary mapping is the fastest because it provides O(1)
average time complexity for lookups, making it more efficient than
multiple conditional checks.")
elif fastest == "if-elif-else":
```

```
    print("If-elif-else is straightforward and efficient for a small
number of conditions, but can become slower with more conditions.")
else:
    print("Match-case provides clear syntax for multiple conditions, but
may have slight overhead compared to dictionary lookups.")
```

**Output:**

C:\Users\ADMIN\AppData\Local\Programs\Python\Python312\python.exe 'c:\Users\ADMIN\Documents\AI - lab\AS-6.3(t2).py'

Enter your rating for the online shopping (1-5): 4
Feedback: Good
Enter your rating for the online shopping (1-5): 2
Feedback: Poor
Enter your rating for the online shopping (1-5): 1
Feedback: Very Poor

Time taken by if-elif-else: 0.0 seconds
Time taken by dictionary mapping: 0.0 seconds
Time taken by match-case: 0.0 seconds

The fastest method is if-elif-else.
If-elif-else is straightforward and efficient for a small number of conditions, but can become slower with more conditions. * Terminal will be reused by tasks, press any key to close it.

**Explanation:**

Clear decision tree.
Readable. Traditional. No tricks