

M.AKHIL REDDY

2303A52315

BATCH 45

Ass 5.5

Task Description #1 (Transparency in Algorithm Optimization)

Task: Use AI to generate two solutions for checking prime numbers:

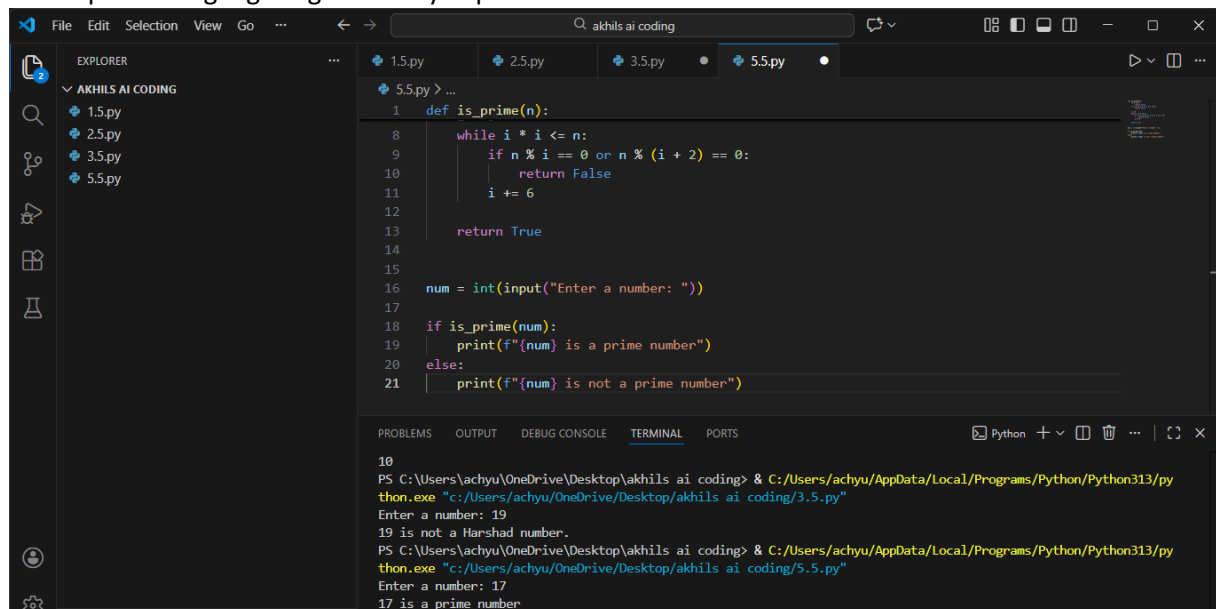
- Naive approach(basic)
- Optimized approach

Prompt:

“Generate Python code for two prime-checking methods and explain how the optimized version improves performance.”

Expected Output:

- Code for both methods.
- Transparent explanation of time complexity.
- Comparison highlighting efficiency improvements



The screenshot shows a Visual Studio Code editor window with a file explorer on the left and a code editor in the center. The file explorer shows a folder named 'AKHILS AI CODING' containing four files: '1.5.py', '2.5.py', '3.5.py', and '5.5.py'. The code editor is open to '5.5.py' and displays the following Python code:

```
1 def is_prime(n):
2     while i * i <= n:
3         if n % i == 0 or n % (i + 2) == 0:
4             return False
5         i += 6
6     return True
7
8 num = int(input("Enter a number: "))
9
10 if is_prime(num):
11     print(f"{num} is a prime number")
12 else:
13     print(f"{num} is not a prime number")
```

Below the code editor is a terminal window showing the execution of the code. The terminal output is as follows:

```
10 PS C:\Users\achyu\OneDrive\Desktop\akhils ai coding> & C:/Users/achyu/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/achyu/OneDrive/Desktop/akhils ai coding/3.5.py"
Enter a number: 19
19 is not a Harshad number.
PS C:\Users\achyu\OneDrive\Desktop\akhils ai coding> & C:/Users/achyu/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/achyu/OneDrive/Desktop/akhils ai coding/5.5.py"
Enter a number: 17
17 is a prime number
```

Task Description #2 (Transparency in Recursive Algorithms)

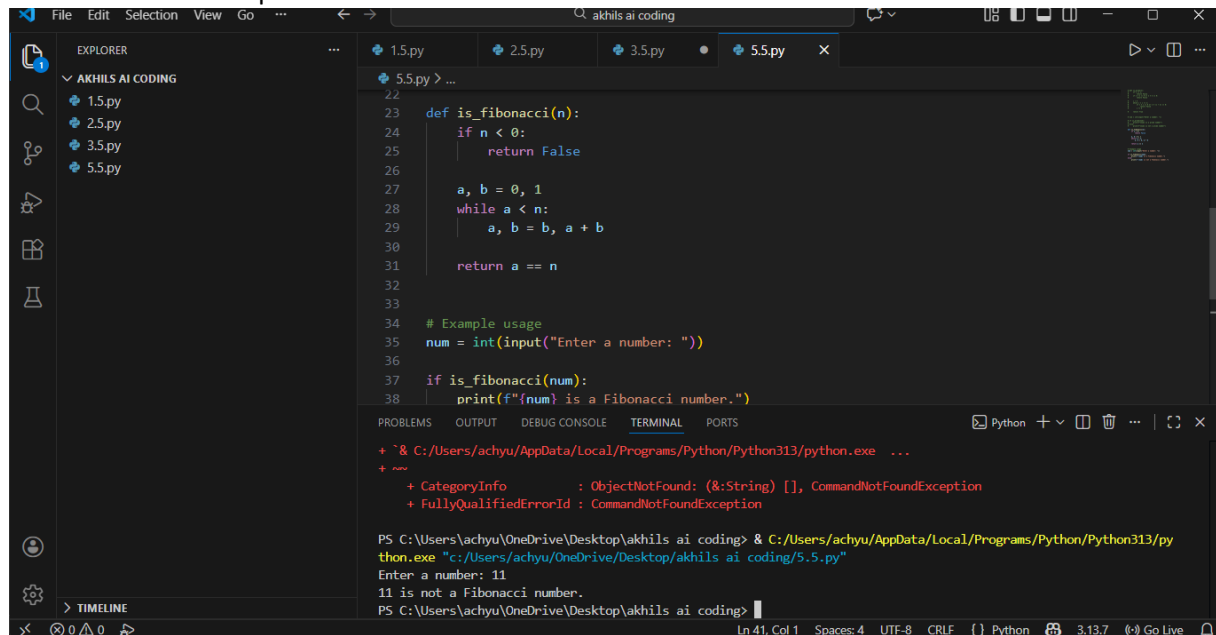
Objective: Use AI to generate a recursive function to calculate Fibonacci numbers.

Instructions:

1. Ask AI to add clear comments explaining recursion.
2. Ask AI to explain base cases and recursive calls.

Expected Output:

- Well-commented recursive code.
- Clear explanation of how recursion works.
- Verification that explanation matches actual execution.



```
22
23 def is_fibonacci(n):
24     if n < 0:
25         return False
26
27     a, b = 0, 1
28     while a < n:
29         a, b = b, a + b
30
31     return a == n
32
33
34 # Example usage
35 num = int(input("Enter a number: "))
36
37 if is_fibonacci(num):
38     print(f"{num} is a Fibonacci number.")
```

```
+ ~
+ CategoryInfo          : ObjectNotFound: (String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException

PS C:\Users\achyu\OneDrive\Desktop\akhils ai coding> & C:/Users/achyu/AppData/Local/Programs/Python/Python313/python.exe ...
PS C:\Users\achyu\OneDrive\Desktop\akhils ai coding> & C:/Users/achyu/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/achyu/OneDrive/Desktop/akhils ai coding/5.5.py"
Enter a number: 11
11 is not a Fibonacci number.
PS C:\Users\achyu\OneDrive\Desktop\akhils ai coding>
```

Use AI to generate a Python program that reads a file and processes data.

Prompt:

“Generate code with proper error handling and clear explanations for each exception.”

Expected Output:

- Code with meaningful exception handling.
- Clear comments explaining each error scenario.
- Validation that explanations align with runtime behavior.

The screenshot shows a Visual Studio Code editor with a project named 'AKHILS AI CODING'. The Explorer sidebar on the left lists four Python files: 1.5.py, 2.5.py, 3.5.py, and 5.5.py. The main editor window displays the code for 5.5.py, which defines a `read_file` function. The function takes a `file_path` as an argument, attempts to open the file, reads its lines, and prints each line. It includes error handling for `FileNotFoundError` and `IOError`. The function is then called with the path `'C:\\Users\\achyu\\Desktop\\sujith hacker\\new.txt'`. Below the editor, the TERMINAL panel shows the command prompt output. It shows the execution of `5.5.py`, an input of `11`, and a message stating `Error: The file at C:\\Users\\achyu\\Desktop\\sujith hacker\\new.txt was not found.`

```
def read_file(file_path):
    try:
        with open(file_path, 'r') as file:
            data = file.readlines()
            # Process data (for example, print each line)
            for line in data:
                print(line.strip())
    except FileNotFoundError:
        print(f"Error: The file at {file_path} was not found.")
    except IOError:
        print("Error: An I/O error occurred while reading the file.")

read_file('C:\\Users\\achyu\\Desktop\\sujith hacker\\new.txt')
```

```
PS C:\Users\achyu\OneDrive\Desktop\akhils ai coding> C:/Users/achyu/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/achyu/OneDrive/Desktop/akhils ai coding/5.5.py"
Enter a number: 11
11 is not a Fibonacci number.
PS C:\Users\achyu\OneDrive\Desktop\akhils ai coding> C:/Users/achyu/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/achyu/OneDrive/Desktop/akhils ai coding/5.5.py"
Error: The file at C:\Users\achyu\Desktop\sujith hacker\new.txt was not found.
PS C:\Users\achyu\OneDrive\Desktop\akhils ai coding>
```

Use an AI tool to generate a Python-based login system.

Analyze: Check whether the AI uses secure password handling practices.

Expected Output:

- Identification of security flaws (plain-text passwords, weak validation).
- Revised version using password hashing and input validation

The screenshot shows the same Visual Studio Code editor with the project 'AKHILS AI CODING'. The Explorer sidebar still lists the four Python files. The main editor window now displays the code for 5.5.py, which defines a `login_system` function. The function takes `stored_username` and `stored_password` as arguments. It prompts the user to enter a username and password, then checks if the entered credentials match the stored ones. If they match, it prints 'Login successful'; otherwise, it prints 'Login failed. Incorrect username or password.' The function is called with `login_system("admin", "password123")`. The TERMINAL panel shows the command prompt output. It shows the execution of `5.5.py`, an input of `admin` for the username and `123` for the password, followed by the message `Login failed. Incorrect username or password.`

```
# generate a code for a login system for password handling
def login_system(stored_username, stored_password):
    username = input("Enter username: ")
    password = input("Enter password: ")

    if username == stored_username and password == stored_password:
        print("Login successful")
    else:
        print("Login failed. Incorrect username or password.")

login_system("admin", "password123")
```

```
11 is not a Fibonacci number.
PS C:\Users\achyu\OneDrive\Desktop\akhils ai coding> C:/Users/achyu/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/achyu/OneDrive/Desktop/akhils ai coding/5.5.py"
Error: The file at C:\Users\achyu\Desktop\sujith hacker\new.txt was not found.
PS C:\Users\achyu\OneDrive\Desktop\akhils ai coding> C:/Users/achyu/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/achyu/OneDrive/Desktop/akhils ai coding/5.5.py"
Enter username: admin
Enter password: 123
Login failed. Incorrect username or password.
PS C:\Users\achyu\OneDrive\Desktop\akhils ai coding>
```

Use an AI tool to generate a Python script that logs user

activity (username, IP address, timestamp).

Analyze: Examine whether sensitive data is logged unnecessarily or insecurely.

Expected Output:

- Identified privacy risks in logging.
- Improved version with minimal, anonymized, or masked logging.
- Explanation of privacy-aware logging principles.

The screenshot displays a Visual Studio Code editor window with a project named 'AKHILS AI CODING'. The Explorer sidebar on the left shows a file named 'user\_activity.log'. The main editor area is open to '5.5.py', which contains two Python functions for logging user activity. The first function, 'log\_user\_activity', logs the username, IP address, and timestamp. The second function, 'log\_user\_activity\_privacy\_aware', logs the username, a masked IP address (e.g., '254.789.xxx.xxx'), and the date. A comment explains that the improved version masks the last two octets of the IP address to protect privacy. The terminal window at the bottom shows the execution of the script using 'python.exe'. It displays an error message about a missing file, followed by the command to run the script. The script prompts for a username ('admin') and password ('123'), and then outputs 'Login failed. Incorrect username or password.'

```
76 def log_user_activity(username, ip_address):
77     logging.basicConfig(filename='user_activity.log', level=logging.INFO)
78     timestamp = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
79     logging.info(f"User: {username}, IP: {ip_address}, Time: {timestamp}")
80
81 # Improved privacy-aware logging (anonymized IP address and date only)
82 def log_user_activity_privacy_aware(username, ip_address):
83     logging.basicConfig(filename='user_activity_privacy.log', level=logging.INFO)
84     date = datetime.now().strftime("%Y-%m-%d")
85     anonymized_ip = ".".join(ip_address.split('.')[:2]) + ".xxx.xxx" # Masking last two
86     logging.info(f"User: {username}, IP: {anonymized_ip}, Date: {date}")
87
88 # Explanation: the improved version masks the last two octets of the IP address to protect
89
90 log_user_activity("user1", "254.789.192.168.1")
91 log_user_activity_privacy_aware("user1", "254.789.192.168.1")
```

thone.exe "c:/Users/achyu/OneDrive/Desktop/akhils ai coding/5.5.py"  
Error: The file at C:\Users\achyu\Desktop\sujith hacker\new.txt was not found.  
PS C:\Users\achyu\OneDrive\Desktop\akhils ai coding> & C:/Users/achyu/AppData/Local/Programs/Python/Python313/py  
thone.exe "c:/Users/achyu/OneDrive/Desktop/akhils ai coding/5.5.py"  
Enter username: admin  
Enter password: 123  
Login failed. Incorrect username or password.  
PS C:\Users\achyu\OneDrive\Desktop\akhils ai coding> & C:/Users/achyu/AppData/Local/Programs/Python/Python313/py  
thone.exe "c:/Users/achyu/OneDrive/Desktop/akhils ai coding/5.5.py"  
PS C:\Users\achyu\OneDrive\Desktop\akhils ai coding> []