INTRODUCTION:

The Python Quiz Project is an interactive console-based application designed to conduct a multiple-choice quiz. The project allows users to answer questions, validates their inputs, and calculates a final score rning and testing knowledge on various subjects in a user-friendly manner. based on their responses. It is an educational tool ideal for lea

METHODOLOGY USED:

The project employs the following methodology:

1. Question Design: A dictionary structure is used to store quiz questions, answer options, and the correct answers.
2. Input Validation: The program ensures valid user input by restoring answers to specific options (A, B, C, D).
3. Answers Evaluation: Used responses are compared to the correct answers stored in the dictionary.
4. Score Calculation: A score counter is incremented for every correct answer, and the total score is displayed at the end of the quiz.
5. Feedback Mechanism: The program provides immediate feedback on whether the selected answer is correct or incorrect.

SOFTWARE REQUIREMENTS:

Python Version: IDLE Python 3.13 (64 bites).

Operating System:

* Windows
* MacOs OR Linux

Editor/IDE: Any Python-supported text editor.

(e.g., PyCharm, VSCode, or Jupyter Notebook).

Dependencies: No external libraries are required; the project uses Python’s built-in modules.

PROGRAM CODE:

# Define questions and answers in a dictionary

quiz\_data = {

"What is the capital of France?": {

"options": ["a) London", "b) Paris", "c) Berlin", "d) Rome"],

"correct": "b"

},

"Who wrote the play 'Romeo and Juliet'?": {

"options": ["a) William Shakespeare", "b) Charles Dickens", "c) Mark Twain", "d) Jane Austen"],

"correct": "a"

},

"What is the chemical symbol for water?": {

"options": ["a) O₂", "b) H₂O", "c) CO₂", "d) N₂O"],

"correct": "b"

},

"Which planet is known as the Red Planet?": {

"options": ["a) Venus", "b) Earth", "c) Mars", "d) Jupiter"],

"correct": "c"

},

"Which of the following is the largest ocean on Earth?": {

"options": ["a) Atlantic Ocean", "b) Indian Ocean", "c) Arctic Ocean", "d) Pacific Ocean"],

"correct": "d"

},

"What is the currency used in Japan?": {

"options": ["a) Yen", "b) Dollar", "c) Euro", "d) Peso"],

"correct": "a"

},

"In which year did the Titanic sink?": {

"options": ["a) 1900", "b) 1912", "c) 1920", "d) 1898"],

"correct": "b"

},

"Who painted the Mona Lisa?": {

"options": ["a) Vincent van Gogh", "b) Leonardo da Vinci", "c) Pablo Picasso", "d) Claude Monet"],

"correct": "b"

},

"What is the largest planet in our solar system?": {

"options": ["a) Earth", "b) Saturn", "c) Jupiter", "d) Neptune"],

"correct": "c"

},

"What is the freezing point of water in Celsius?": {

"options": ["a) 0°C", "b) 32°F", "c) 100°C", "d) -32°C"],

"correct": "a"

}

}

# Initialize variables

score = 0

total\_questions = len(quiz\_data)

# Loop through each question in the quiz

for question, data in quiz\_data.items():

print(f"\n{question}")

# Display the options

for option in data["options"]:

print(option)

# Input validation loop

answer = ""

while answer not in ['a', 'b', 'c', 'd']:

answer = input("Please select your answer (a, b, c, or d): ").lower()

if answer not in ['a', 'b', 'c', 'd']:

print("Invalid input. Please enter a valid option (a, b, c, or d).")

# Check if the answer is correct

if answer == data["correct"]:

score += 1

print("Correct!")

else:

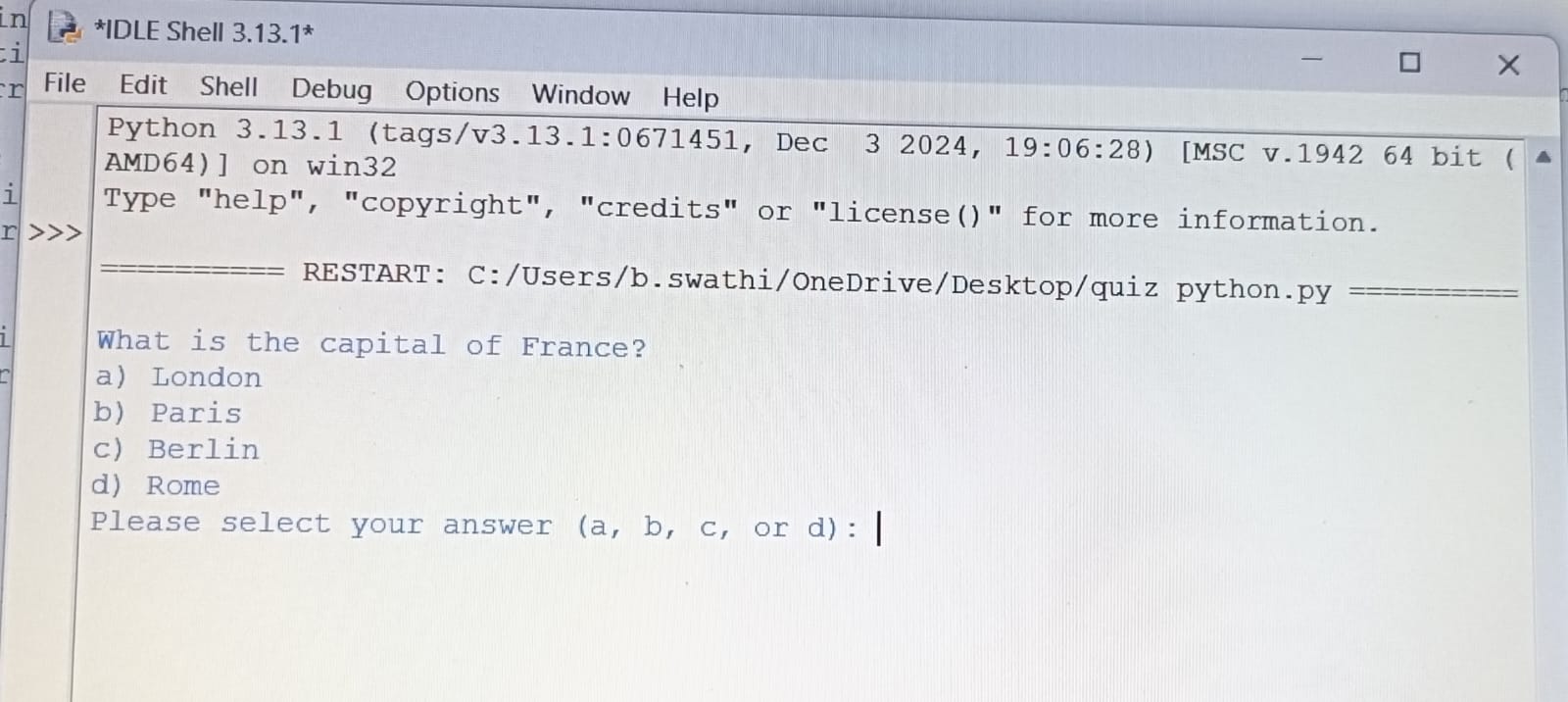
print("Incorrect.")

# Display the final score

print(f"\nYour final score is {score}/{total\_questions}.")

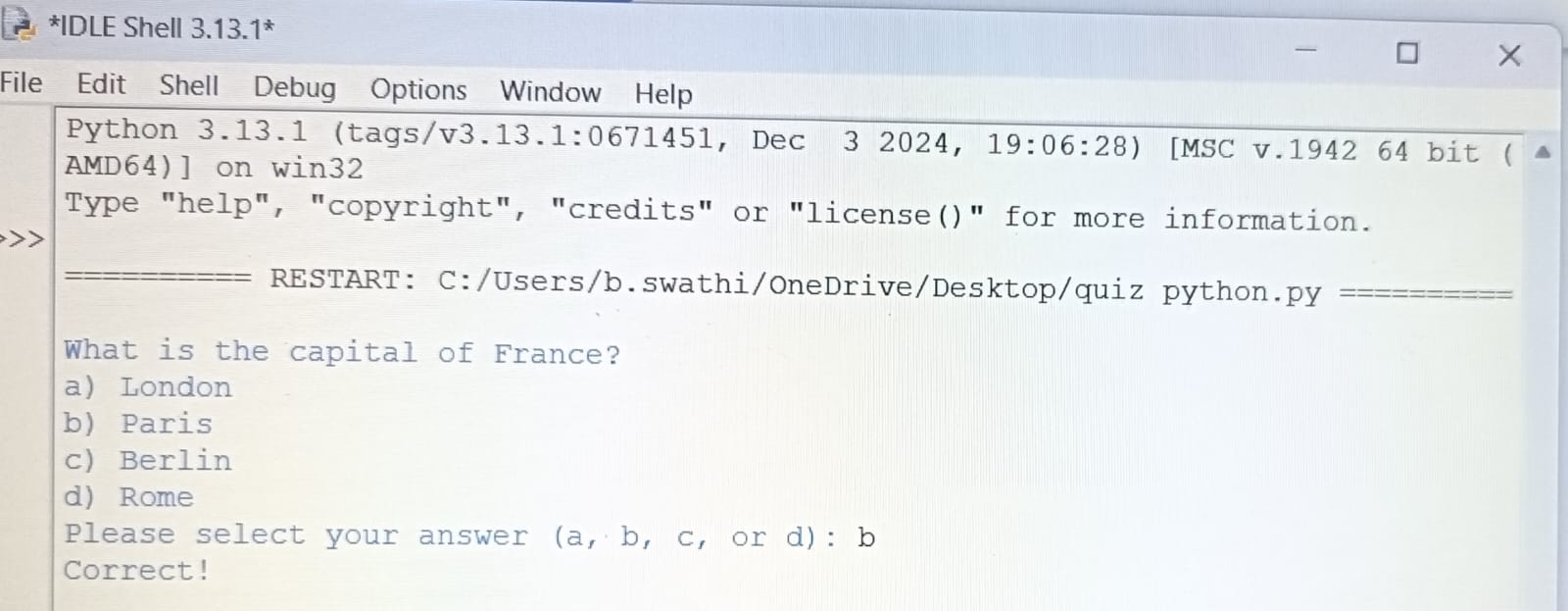
OUTPUT SCREENSHORTS:

* Initial Quention prompt: Display the question and options to the user.

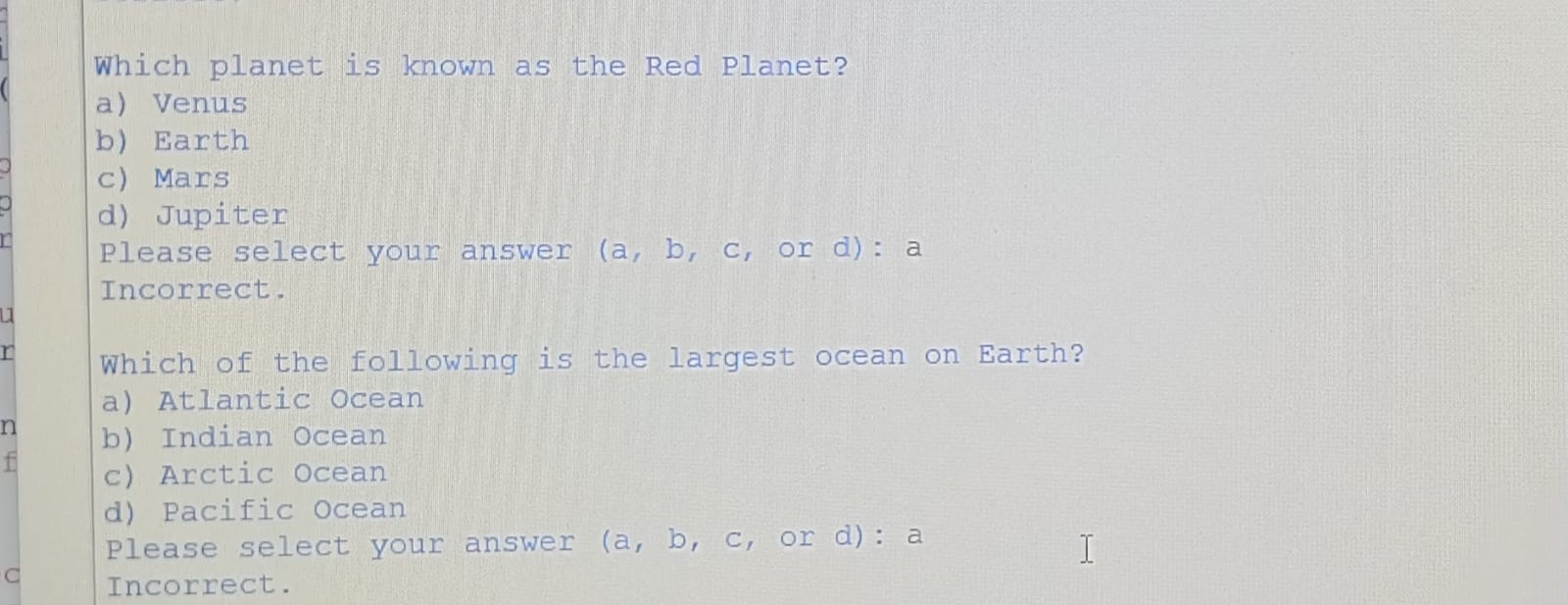


* Feedback on Answer: Provides feedback based on the user’s input, indicating whether the answer is correct or incorrect.

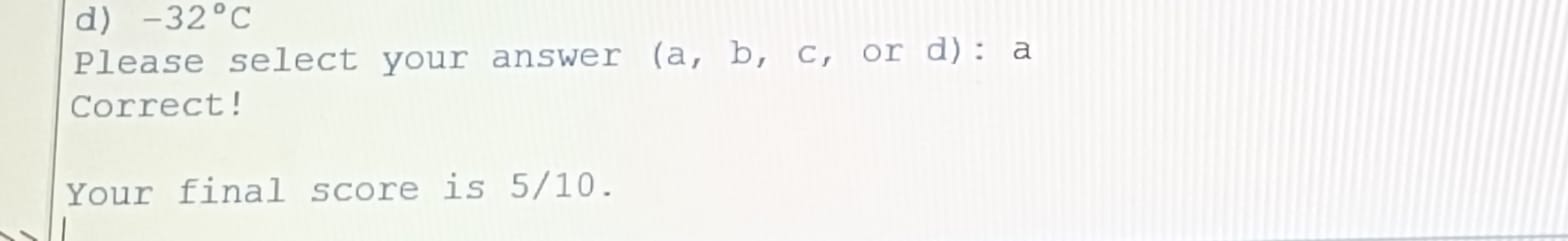
For correct answer:



For incorrect answer:



* Final Score: A Summary screen showing the user’s total score out of the number of questions.



FURTHER OUTCOMES:

* GUI Implementation: Enhancing the project with a graphical user interface (GUI) using libraries like Tkinter or PyQt for better interactivity.
* Question Bank Integration: Expanding the quiz by linking it to an external database or file system for dynamic question loading.
* Online Accessibility: Transforming the project into a web-based quiz applications using framewirks like Flask or Django.
* Real-Time Leaderboard: Adding a feature to maintain user scores and display leaderboards for competitive quizzes.

CONCLUSION:

The Python Quiz Project demonstrates the practical application of Python programming for educational purposes. It is a lightweight, efficient, and user- friendly tool that can be further enhanced with additional features for broader usability.