2303A51563

batch=10

Task 1: Zero-Shot Prompting – Leap Year Check

Scenario

Zero-shot prompting involves giving instructions without providing examples.

Task Description

Use zero-shot prompting to instruct an AI tool to generate a Python function that:

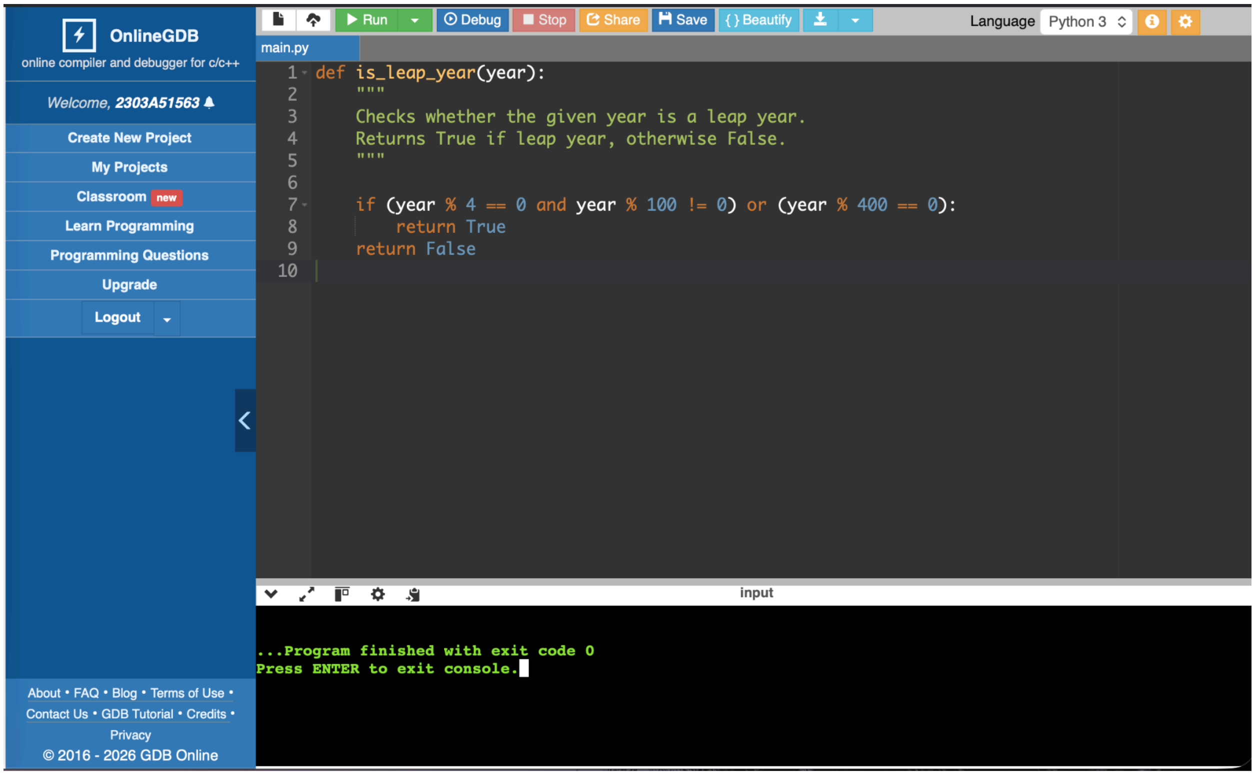
- Accepts a year as input
- Checks whether the given year is a leap year
- Returns an appropriate result

Note: No input-output examples should be provided in the prompt.

Expected Output

- AI-generated leap year checking function
- Correct logical conditions
- Sample input and output
- Screenshot of AI-generated response (if required)

```
def is_leap_year(year):
    """
    Checks whether the given year is a leap year.
    Returns True if leap year, otherwise False.
code
    """
    if (year % 4 == 0 and year % 100 != 0) or (year % 400
== 0):
        return True
    return False
```



```python
def is_leap_year(year):
    """
    Checks whether the given year is a leap year.
    Returns True if leap year, otherwise False.
    """

    if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):
        return True
    return False
```

```
...Program finished with exit code 0
Press ENTER to exit console.
```

ask 2: One-Shot Prompting – Centimeters to Inches Conversion

Scenario

One-shot prompting guides AI using a single example.

Task Description

Use one-shot prompting by providing one input-output example to generate a Python function that:

- Converts centimeters to inches
- Uses the correct mathematical formula

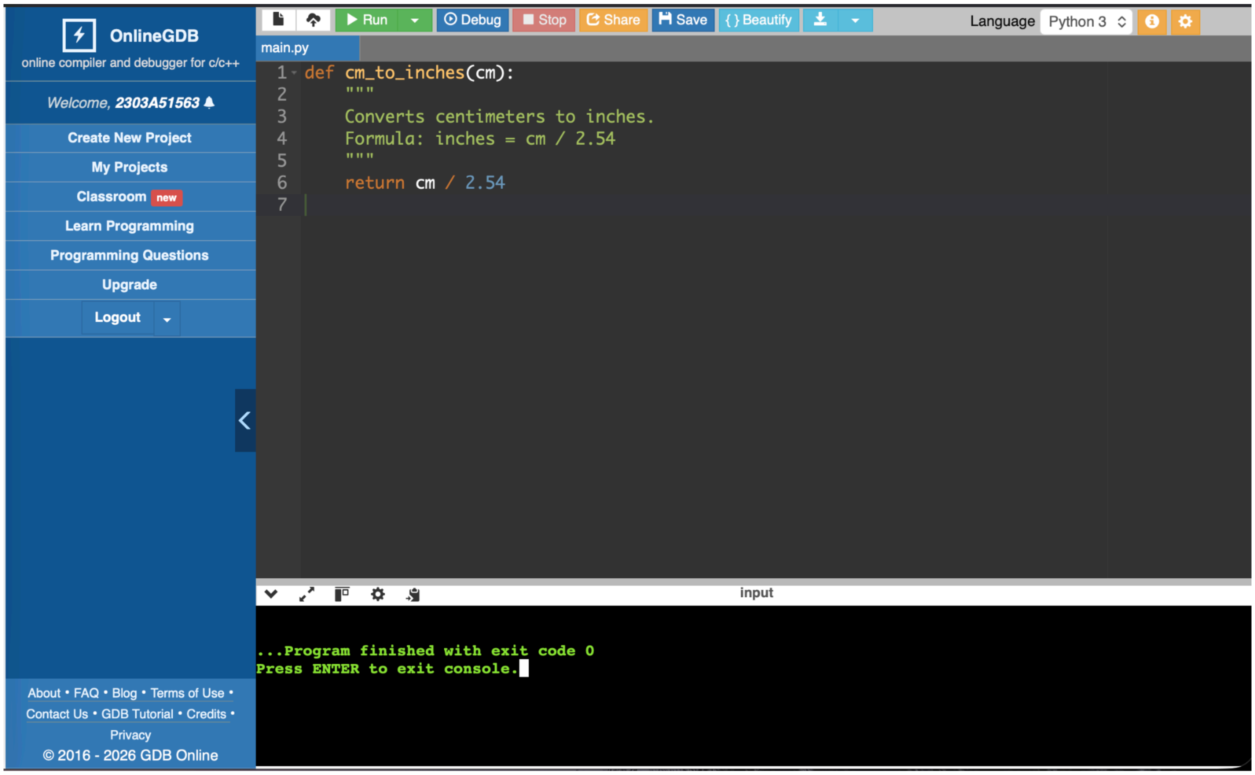
Example provided in prompt:

Input: 10 cm → Output: 3.94 inches

Expected Output

- Python function with correct conversion logic
- Accurate calculation
- Sample test cases and outputs

Code



```
def cm_to_inches(cm):
    """
    Converts centimeters to inches.
    Formula: inches = cm / 2.54
    """
    return cm / 2.54

```

```
...Program finished with exit code 0
Press ENTER to exit console.
```

```
def cm_to_inches(cm):
    """
    Converts centimeters to inches.
    Formula: inches = cm / 2.54
    """
    return cm / 2.54
```

Scenario

Few-shot prompting improves accuracy by providing multiple examples.

Task Description

Use few-shot prompting with 2–3 examples to generate a Python function that:

- Accepts a full name as input
- Formats it as “Last, First”

Example formats:

- "John Smith" → "Smith, John"
- "Anita Rao" → "Rao, Anita"

Expected Output

- Well-structured Python function
- Output strictly following example patterns
- Correct handling of names
- Sample inputs and outputs

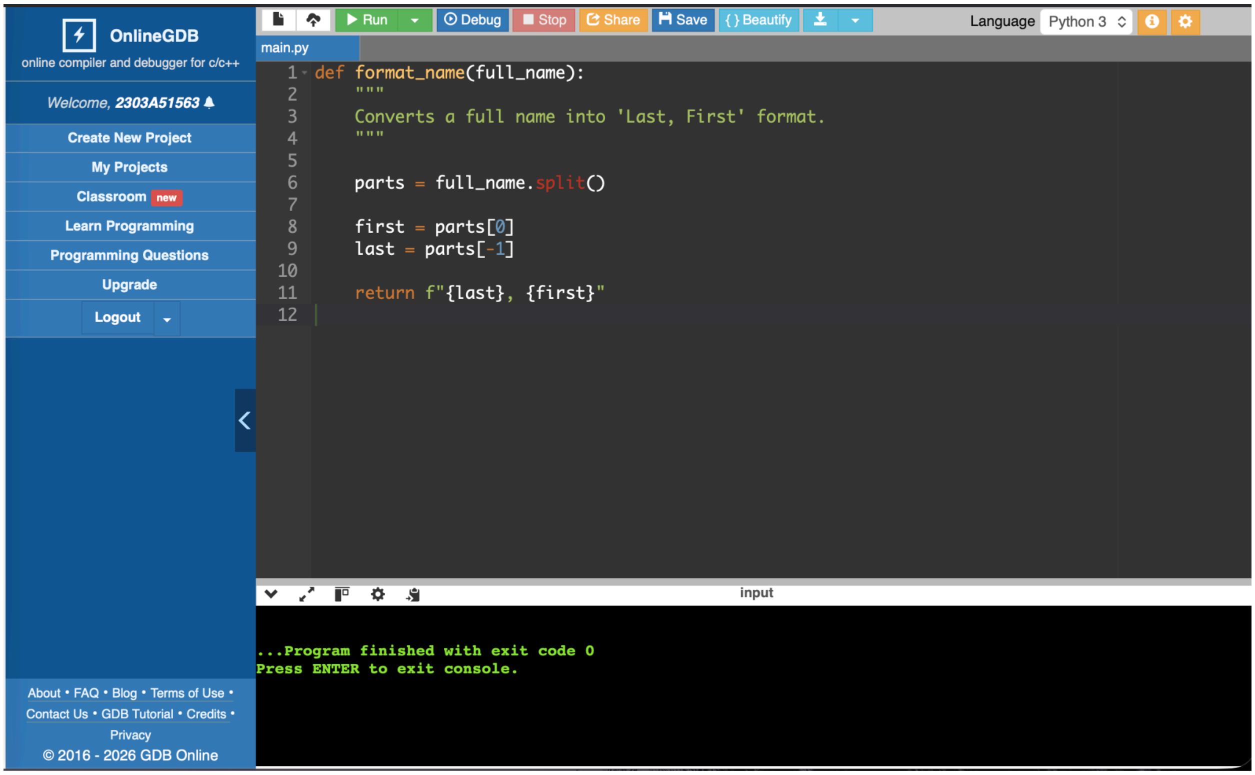
Code

```
def format_name(full_name):
    """
    Converts a full name into 'Last, First' format.
    """


    parts = full_name.split()

    first = parts[0]
    last = parts[-1]

    return f"{last}, {first}"
```



```python
def format_name(full_name):
    """
    Converts a full name into 'Last, First' format.
    """

    parts = full_name.split()

    first = parts[0]
    last = parts[-1]

    return f"{last}, {first}"
```

```
...Program finished with exit code 0
Press ENTER to exit console.
```

ask 4: Comparative Analysis – Zero-Shot vs Few-Shot Scenario

Different prompt strategies may produce different code quality.

Task Description

- Use zero-shot prompting to generate a function that counts vowels in a string
- Use few-shot prompting for the same problem
- Compare both outputs based on:

o Accuracy

o Readability

o Logical clarity

Expected Output

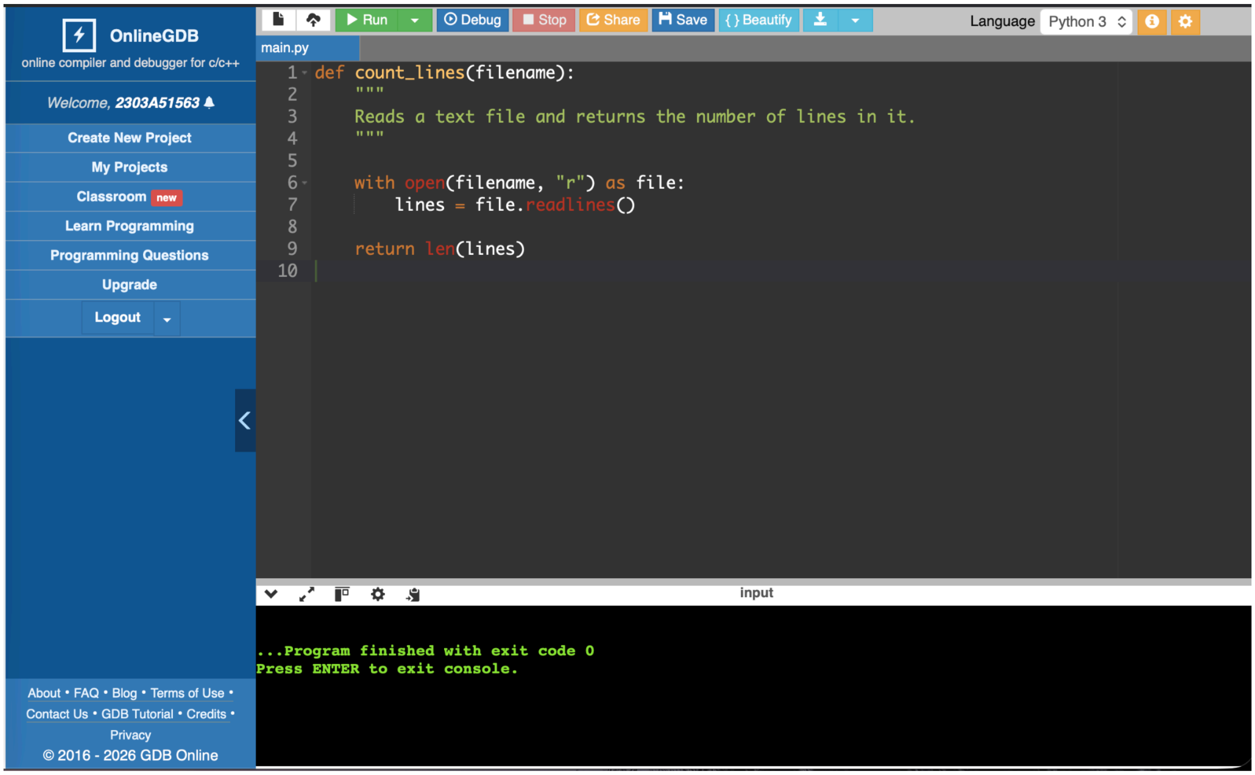
- Two vowel-counting functions
- Comparison table or short reflection paragraph
- Conclusion on prompt effectiveness

Code

```
def count_lines(filename):
    """
    Reads a text file and returns the number of lines in it.
    """

    with open(filename, "r") as file:
        lines = file.readlines()

    return len(lines)
```



```python
def count_lines(filename):
    """
    Reads a text file and returns the number of lines in it.
    """

    with open(filename, "r") as file:
        lines = file.readlines()

    return len(lines)

```

```
...Program finished with exit code 0
Press ENTER to exit console.
```