

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.

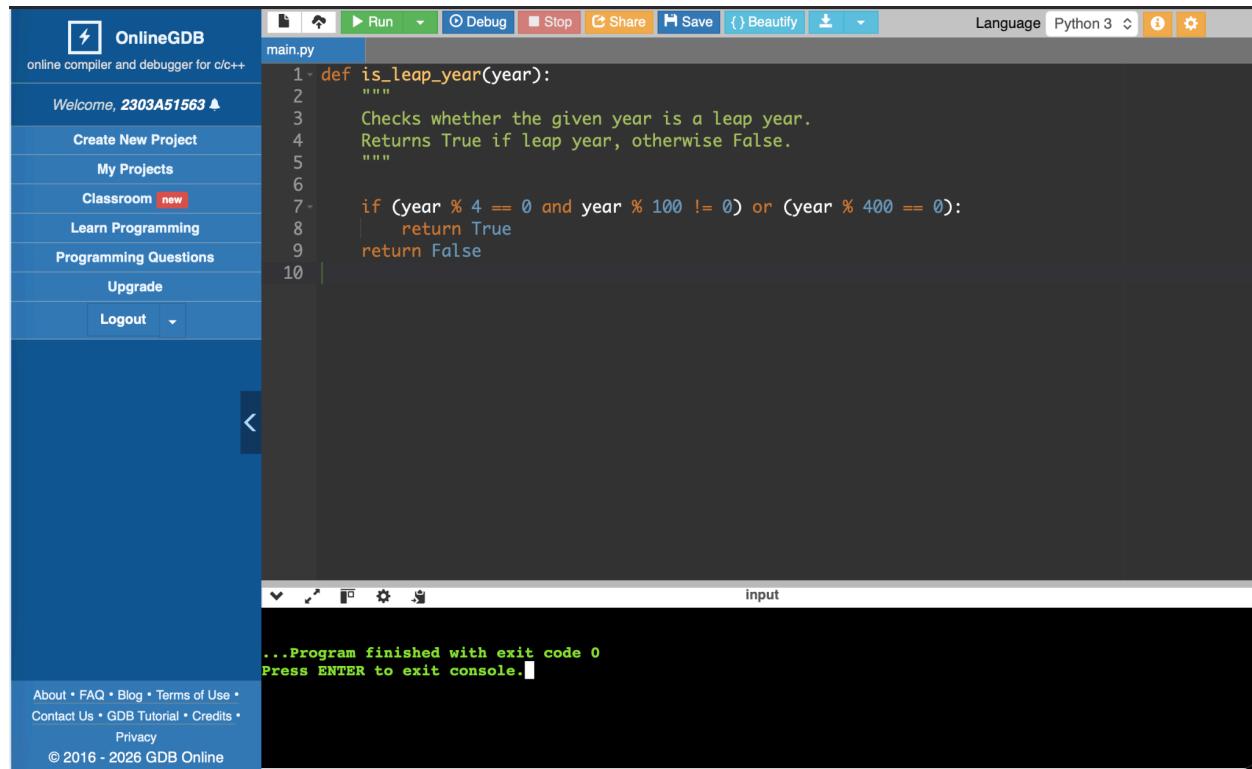
code

"""

```

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

```



The screenshot shows the OnlineGDB interface. On the left, there's a sidebar with various options like 'Create New Project', 'My Projects', 'Classroom', 'Learn Programming', 'Programming Questions', 'Upgrade', and 'Logout'. The main area has a dark theme with a toolbar at the top featuring 'Run', 'Debug', 'Stop', 'Save', and 'Beautify' buttons. The language is set to 'Python 3'. A code editor window titled 'main.py' contains the following Python code:

```

1 def is_leap_year(year):
2     """
3         Checks whether the given year is a leap year.
4         Returns True if leap year, otherwise False.
5     """
6
7     if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):
8         return True
9     return False
10

```

Below the code editor is a terminal window showing the output of the program:

```

...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

The screenshot shows the OnlineGDB interface. On the left, there's a sidebar with user information ('Welcome, 2303A51563'), project options ('Create New Project', 'My Projects', 'Classroom'), learning resources ('Learn Programming'), and account management ('Upgrade', 'Logout'). The main area has tabs for 'main.py' (selected), 'main.c', and 'main.cpp'. The 'main.py' tab contains the following code:

```

1 def cm_to_inches(cm):
2     """
3         Converts centimeters to inches.
4         Formula: inches = cm / 2.54
5     """
6     return cm / 2.54
7

```

Below the code editor is a terminal window titled 'Input' which shows the output of the program:

```

...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.
```

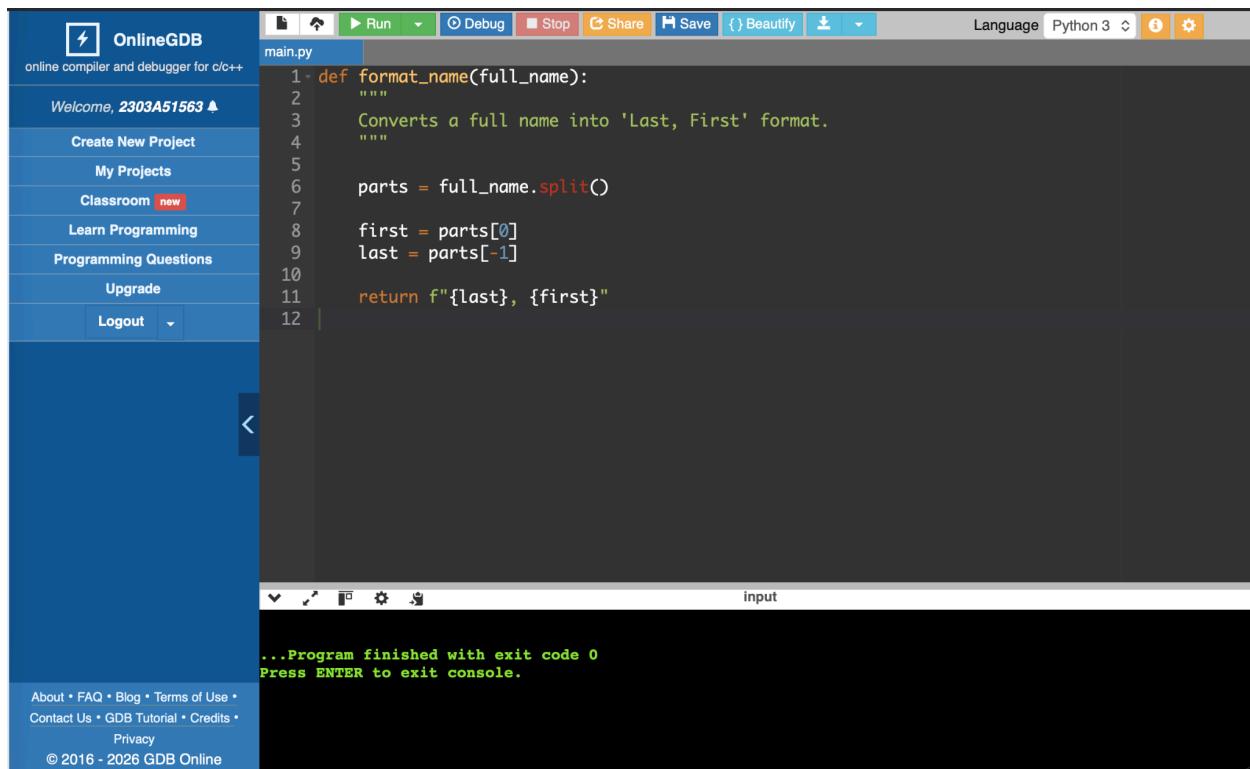
```
    """
```

```
    parts = full_name.split()
```

```
    first = parts[0]
```

```
last = parts[-1]
```

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



The screenshot shows the OnlineGDB interface. On the left, there's a sidebar with navigation links like 'Create New Project', 'My Projects', 'Classroom', 'Learn Programming', 'Programming Questions', 'Upgrade', and 'Logout'. The main area has tabs for 'main.py' and 'input'. The code in 'main.py' is:

```
1 def format_name(full_name):
2     """
3         Converts a full name into 'Last, First' format.
4     """
5
6     parts = full_name.split()
7
8     first = parts[0]
9     last = parts[-1]
10
11    return f"{last}, {first}"
```

The 'input' tab shows the output of the program:

```
...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:
 - 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)
```

The screenshot shows the OnlineGDB interface. The top navigation bar includes 'Run', 'Debug', 'Stop', 'Share', 'Save', 'Beautify', and language selection for 'Python 3'. The left sidebar has links for 'Create New Project', 'My Projects', 'Classroom', 'Learn Programming', 'Programming Questions', 'Upgrade', and 'Logout'. The main workspace displays a Python script named 'main.py' with the following code:

```
1 def count_lines(filename):
2     """
3         Reads a text file and returns the number of lines in it.
4     """
5
6     with open(filename, "r") as file:
7         lines = file.readlines()
8
9     return len(lines)
10
```

The bottom console window shows the output of the program execution:

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