

S 1 Z ro-shot: Leap Year

```
C: > Users > aakar > OneDrive > Documents > Online gdb > task_1.py > ...
1  def is_leap_year(year: int) -> bool:
2      return (year % 4 == 0) and (year % 100 != 0 or year % 400 == 0)
3
4  # User input
5  year = int(input("Enter a year: "))
6  print(f"Input: {year} → Output: {is_leap_year(year)}")
7
```

Output :

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Enter a year: 2024

Input: 2024 → Output: True

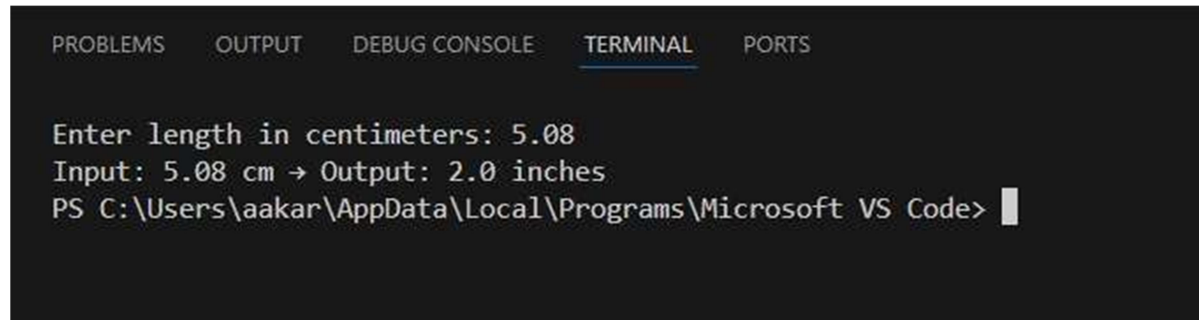
PS C:\Users\aaakar\AppData\Local\Programs\Microsoft VS Code> █

2 One-shot: cm → inches

```
C: > Users > aakar > OneDrive > Documents > Online gdb > task_1.py > ...
1  def cm_to_inches(cm: float) -> float:
2      return cm / 2.54
3
4  # User input
5  cm = float(input("Enter length in centimeters: "))
6  print(f"Input: {cm} cm → Output: {cm_to_inches(cm)} inches")
7
```

Task — e

Output :



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Enter length in centimeters: 5.08
Input: 5.08 cm → Output: 2.0 inches
PS C:\Users\aaakar\AppData\Local\Programs\Microsoft VS Code> |
```

The image shows a screenshot of a terminal window within a code editor. The terminal has a dark background with light-colored text. At the top, there is a tab bar with five tabs: 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL' (which is selected and underlined), and 'PORTS'. Below the tabs, the terminal displays the following text: 'Enter length in centimeters: 5.08', 'Input: 5.08 cm → Output: 2.0 inches', and a command prompt 'PS C:\Users\aaakar\AppData\Local\Programs\Microsoft VS Code>' followed by a cursor.

Task — w

3 F -shot: Format name

```
C: > Users > aakar > OneDrive > Documents > Online gdb > task_1.py > ...
1  def format_name_last_first(full_name: str) -> str:
2      parts = [p for p in full_name.strip().split() if p]
3      if not parts:
4          return ""
5      if len(parts) == 1:
6          return parts[0]
7      last = parts[-1]
8      first_block = " ".join(parts[:-1])
9      return f"{last}, {first_block}"
10
11 # User input
12 name = input("Enter full name: ")
13 print(f"Input: {name} → Output: {format_name_last_first(name)}")
14
```

Output :

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
Enter full name: John Doe
Input: John Doe → Output: Doe, John
```

```
PS C:\Users\aarar\AppData\Local\Programs\Microsoft VS Code> █
```

4 Vo el Counter

```
C: > Users > aakar > OneDrive > Documents > Online gdb > task_1.py > ...
1  def count_vowels(s: str, *, include_y: bool = False) -> int:
2      vowels = set("aeiou" + ("y" if include_y else ""))
3      return sum(ch.lower() in vowels for ch in s)
4
5  # User input
6  text = input("Enter a string: ")
7  print(f"Input: {text} → Output: {count_vowels(text)} vowels")
8
```

Task — ew

Output :

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Enter a string: qwertyuiop
Input: qwertyuiop → Output: 4 vowels
PS C:\Users\aaakar\AppData\Local\Programs\Microsoft VS Code> █
```

5 F -shot: Count lines in file

```
C: > Users > aaakar > OneDrive > Documents > Online gdb > task_1.py > ...
1  from pathlib import Path
2
3  def count_lines_in_file(path: str | Path) -> int:
4      p = Path(path)
5      if not p.exists():
6          raise FileNotFoundError(f"No such file: {p}")
7      with p.open("r", encoding="utf-8", errors="replace") as f:
8          return sum(1 for _ in f)
9
10 # User input
11 file_name = input("Enter file name: ")
12 print(f"Input: {file_name} → Output: {count_lines_in_file(file_name)} lines")
13
```

Output :

```
Enter file name: qwerty1
Traceback (most recent call last):
  File "c:\Users\aaakar\OneDrive\Documents\Online gdb\task_1.py", line 12, in <module>
    print(f"Input: {file_name} → Output: {count_lines_in_file(file_name)} lines")
                                             ~~~~~^~~~~~
  File "c:\Users\aaakar\OneDrive\Documents\Online gdb\task_1.py", line 6, in count_lines_in_file
    raise FileNotFoundError(f"No such file: {p}")
FileNotFoundError: No such file: qwerty1
PS C:\Users\aaakar\AppData\Local\Programs\Microsoft VS Code> █
```