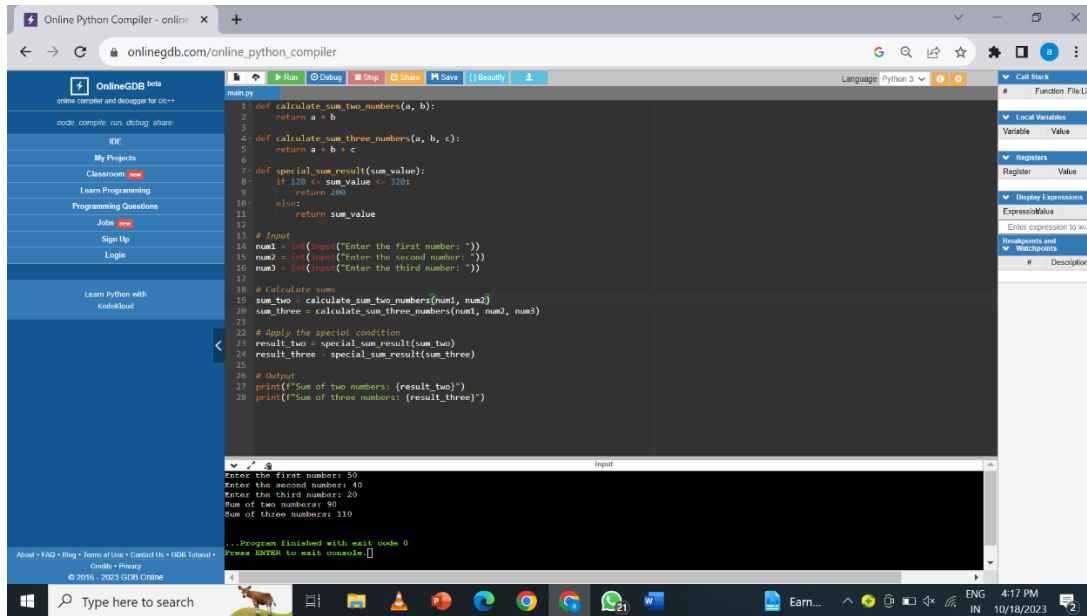


1. Write a python program to calculate the sum of Two numbers and Three numbers.
However, if the sum is between 120 to 320 it will return 200.

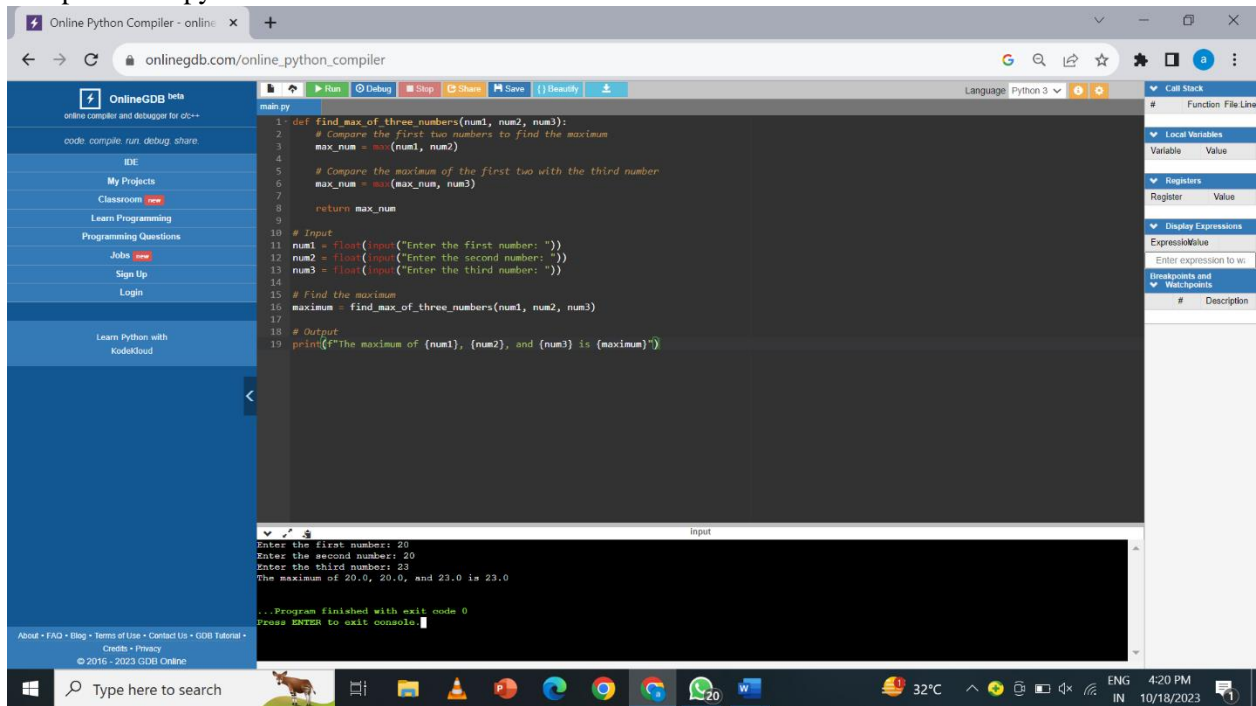


The screenshot shows the Online Python Compiler interface. The code in the editor is as follows:

```
1 def calculate_sum_two_numbers(a, b):  
2     return a + b  
3  
4 def calculate_sum_three_numbers(a, b, c):  
5     return a + b + c  
6  
7 def special_sum_result(sum_value):  
8     if 120 < sum_value <= 320:  
9         return 200  
10    else:  
11        return sum_value  
12  
13 # Input  
14 num1 = int(input("Enter the first number: "))  
15 num2 = int(input("Enter the second number: "))  
16 num3 = int(input("Enter the third number: "))  
17  
18 # Calculate sums  
19 sum_two = calculate_sum_two_numbers(num1, num2)  
20 sum_three = calculate_sum_three_numbers(num1, num2, num3)  
21  
22 # Apply the special condition  
23 result_two = special_sum_result(sum_two)  
24 result_three = special_sum_result(sum_three)  
25  
26 # Output  
27 print(f"Sum of two numbers: {result_two}")  
28 print(f"Sum of three numbers: {result_three}")
```

The console output shows the program execution with inputs 50, 40, and 20, resulting in sums of 90 and 110.

2. Implement a python function to find the Maximum of Three numbers

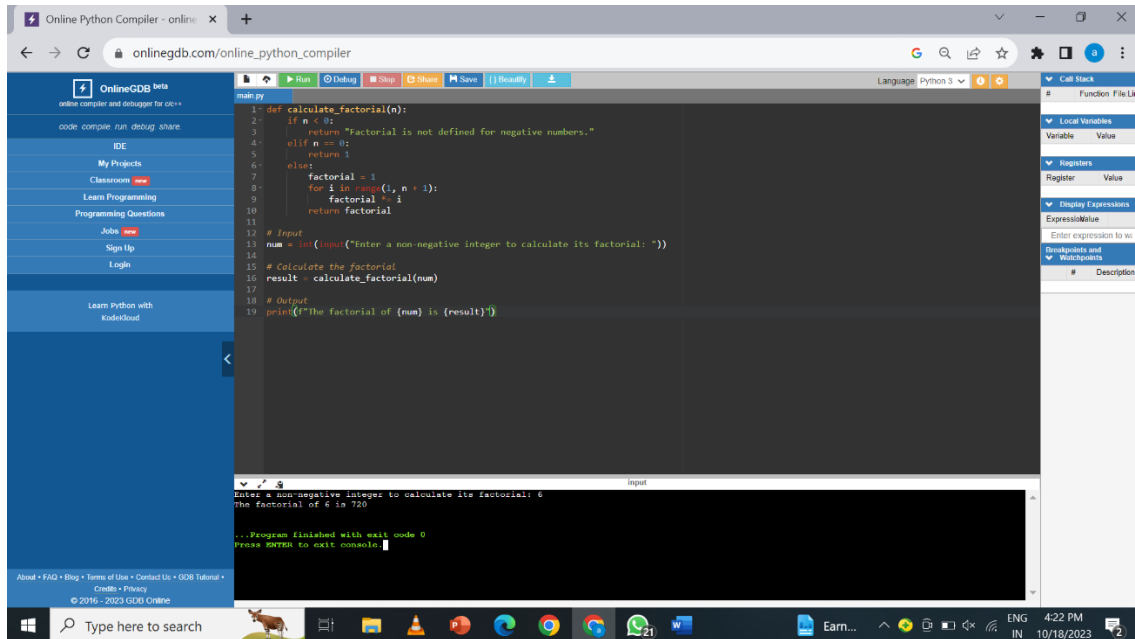


The screenshot shows the Online Python Compiler interface. The code in the editor is as follows:

```
1 def find_max_of_three_numbers(num1, num2, num3):  
2     # Compare the first two numbers to find the maximum  
3     max_num = max(num1, num2)  
4  
5     # Compare the maximum of the first two with the third number  
6     max_num = max(max_num, num3)  
7  
8     return max_num  
9  
10 # Input  
11 num1 = float(input("Enter the first number: "))  
12 num2 = float(input("Enter the second number: "))  
13 num3 = float(input("Enter the third number: "))  
14  
15 # Find the maximum  
16 maximum = find_max_of_three_numbers(num1, num2, num3)  
17  
18 # Output  
19 print(f"The maximum of {num1}, {num2}, and {num3} is {maximum}")
```

The console output shows the program execution with inputs 20, 20, and 23, resulting in a maximum value of 23.0.

3. Write a python program to calculate the Factorial of a given number.



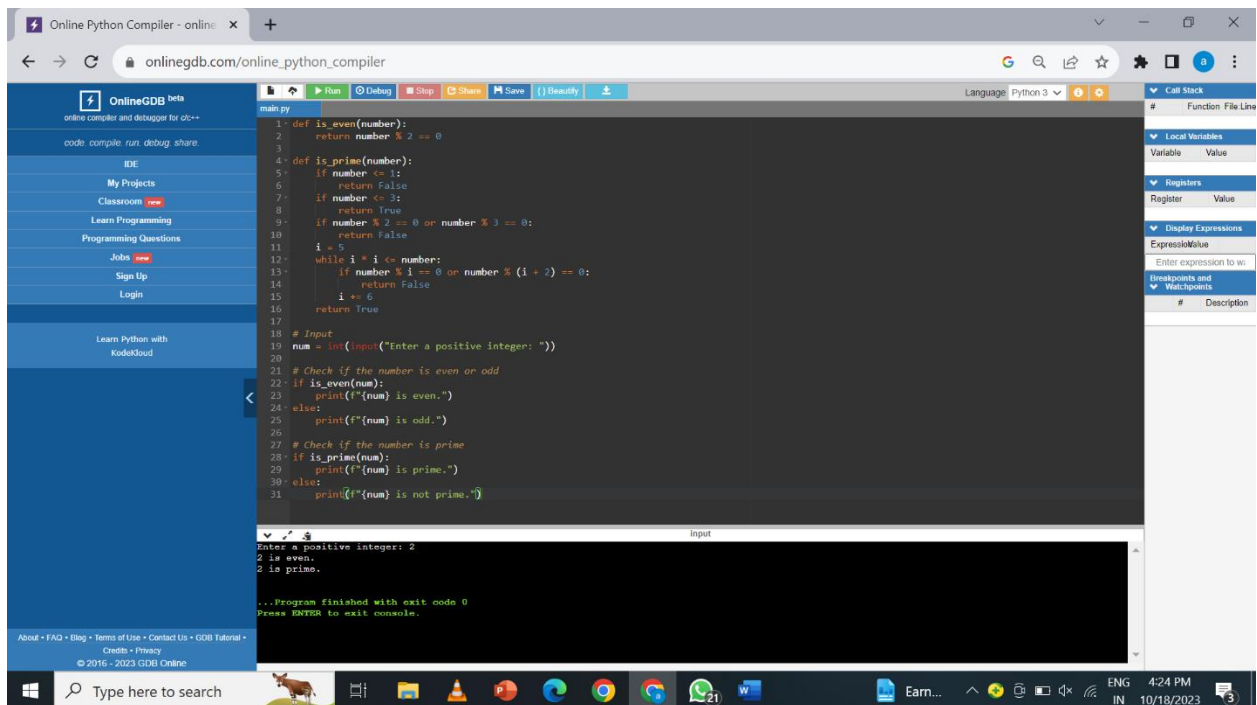
The screenshot shows the Online Python Compiler interface. The code editor contains a Python program to calculate the factorial of a given number. The program defines a function `calculate_factorial(n)` that returns the factorial of `n`. It then takes user input, calculates the factorial, and prints the result.

```
1 def calculate_factorial(n):
2     if n < 0:
3         return "Factorial is not defined for negative numbers."
4     elif n == 0:
5         return 1
6     else:
7         factorial = 1
8         for i in range(1, n + 1):
9             factorial *= i
10        return factorial
11
12 # Input
13 num = int(input("Enter a non-negative integer to calculate its factorial: "))
14
15 # Calculate the factorial
16 result = calculate_factorial(num)
17
18 # Output
19 print(f"The factorial of {num} is {result}")
```

The console output shows the program execution for input 4, resulting in the factorial 24.

```
Enter a non-negative integer to calculate its factorial: 4
The factorial of 4 is 24
...Program finished with exit code 0
Press ENTER to exit console.
```

4. Write a python program to Check if a Number is Even or Odd and also check whether it is Prime or not.



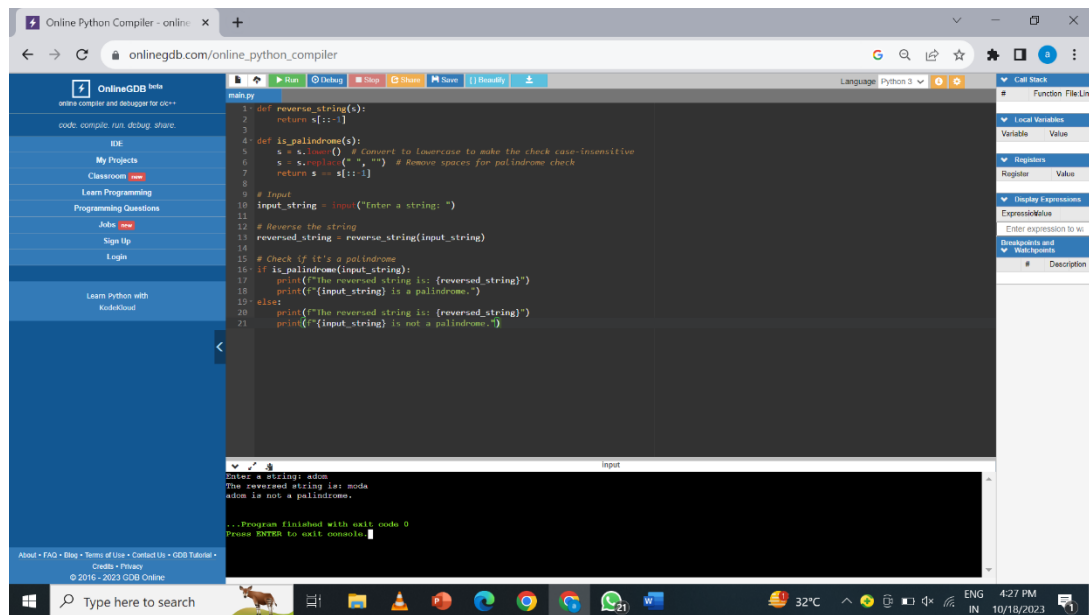
The screenshot shows the Online Python Compiler interface. The code editor contains a Python program that checks if a number is even or odd and also checks if it is prime. The program defines functions `is_even(number)` and `is_prime(number)`. It then takes user input, checks the conditions, and prints the results.

```
1 def is_even(number):
2     return number % 2 == 0
3
4 def is_prime(number):
5     if number <= 1:
6         return False
7     if number <= 3:
8         return True
9     if number % 2 == 0 or number % 3 == 0:
10        return False
11    i = 5
12    while i * i <= number:
13        if number % i == 0 or number % (i + 2) == 0:
14            return False
15        i += 6
16    return True
17
18 # Input
19 num = int(input("Enter a positive integer: "))
20
21 # Check if the number is even or odd
22 if is_even(num):
23     print(f"{num} is even.")
24 else:
25     print(f"{num} is odd.")
26
27 # Check if the number is prime
28 if is_prime(num):
29     print(f"{num} is prime.")
30 else:
31     print(f"{num} is not prime.")
```

The console output shows the program execution for input 2, resulting in the output "2 is even." and "2 is prime."

```
Enter a positive integer: 2
2 is even.
2 is prime.
...Program finished with exit code 0
Press ENTER to exit console.
```

5. Implement a python function to Reverse a given String and also check for palindrome or not

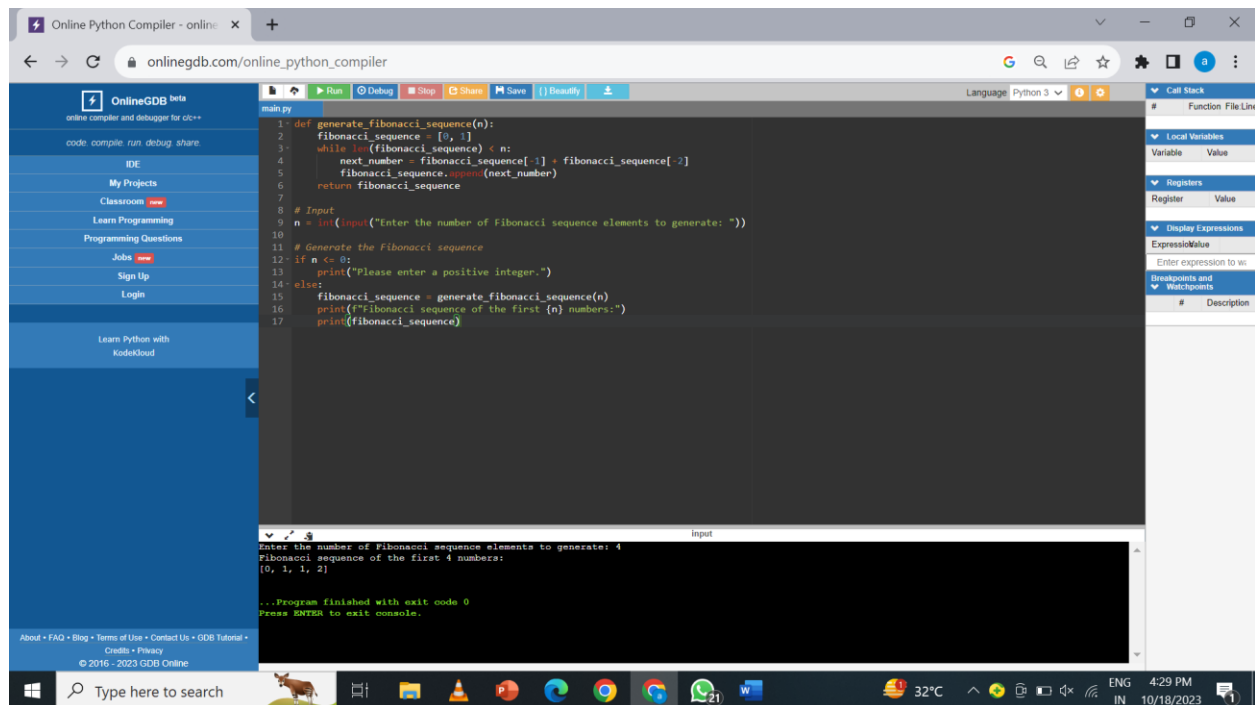


The screenshot shows the Online Python Compiler interface. The code in the main editor is as follows:

```
1 def reverse_string(s):  
2     return s[::-1]  
3  
4 def is_palindrome(s):  
5     s = s.lower() # Convert to lowercase to make the check case-insensitive  
6     s = s.replace(" ", "") # Remove spaces for palindrome check  
7     return s == s[::-1]  
8  
9 # Input  
10 input_string = input("Enter a string: ")  
11  
12 # Reverse the string  
13 reversed_string = reverse_string(input_string)  
14  
15 # Check if it's a palindrome  
16 if is_palindrome(input_string):  
17     print(f"The reversed string is: {reversed_string}")  
18     print(f"{input_string} is a palindrome.")  
19 else:  
20     print(f"The reversed string is: {reversed_string}")  
21     print(f"{input_string} is not a palindrome.")
```

The console output shows the program execution with the input "adon" and the output "The reversed string is: noda" and "adon is not a palindrome." The program finished with exit code 0.

6. Write a python program to Generate Fibonacci Sequence.



The screenshot shows the Online Python Compiler interface. The code in the main editor is as follows:

```
1 def generate_fibonacci_sequence(n):  
2     fibonacci_sequence = [0, 1]  
3     while len(fibonacci_sequence) < n:  
4         next_number = fibonacci_sequence[-1] + fibonacci_sequence[-2]  
5         fibonacci_sequence.append(next_number)  
6     return fibonacci_sequence  
7  
8 # Input  
9 n = int(input("Enter the number of Fibonacci sequence elements to generate: "))  
10  
11 # Generate the Fibonacci sequence  
12 if n <= 0:  
13     print("Please enter a positive integer.")  
14 else:  
15     fibonacci_sequence = generate_fibonacci_sequence(n)  
16     print(f"Fibonacci sequence of the first {n} numbers:")  
17     print(fibonacci_sequence)
```

The console output shows the program execution with the input "4" and the output "Fibonacci sequence of the first 4 numbers: [0, 1, 1, 2]". The program finished with exit code 0.

7. Write a python program to calculate the area and perimeter of different geometric shapes (circle, rectangle, triangle, etc.).

The image displays two screenshots of an online Python compiler interface, specifically the OnlineGDB website. The interface includes a sidebar with navigation links, a central code editor, and a right-hand panel for debugging tools.

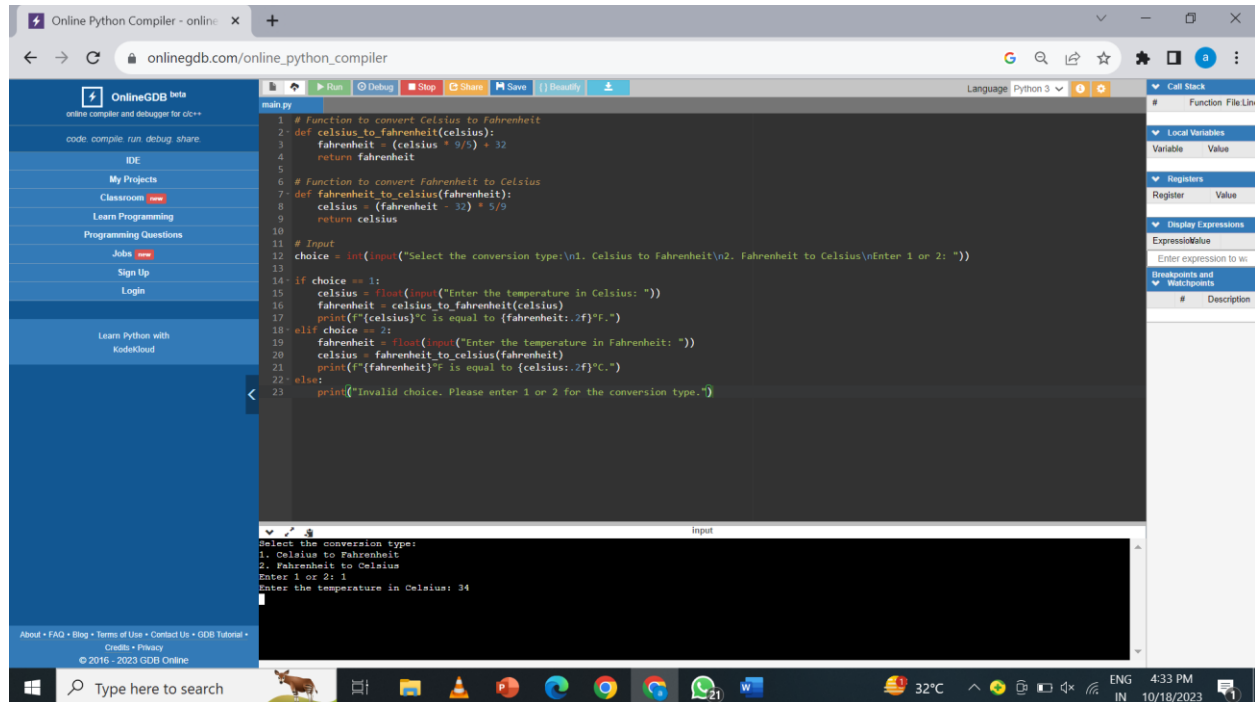
First Screenshot: The code editor shows a Python program that defines functions for calculating the area and perimeter of a circle, a rectangle, and a triangle. The program uses a menu-driven approach to select the shape and calculate its area and perimeter. The code is as follows:

```
1 import math
2
3 # Function to calculate the area and perimeter of a circle
4 def calculate_circle_area_and_perimeter(radius):
5     area = math.pi * radius ** 2
6     perimeter = 2 * math.pi * radius
7     return area, perimeter
8
9 # Function to calculate the area and perimeter of a rectangle
10 def calculate_rectangle_area_and_perimeter(length, width):
11     area = length * width
12     perimeter = 2 * (length + width)
13     return area, perimeter
14
15 # Function to calculate the area and perimeter of a triangle
16 def calculate_triangle_area_and_perimeter(base, height, side1, side2, side3):
17     perimeter = side1 + side2 + side3
18     s = perimeter / 2 # Semi-perimeter for Heron's formula
19     area = math.sqrt(s * (s - side1) * (s - side2) * (s - side3))
20     return area, perimeter
21
22 # Menu for shape selection
23 print("Select a geometric shape:")
24 print("1. Circle")
25 print("2. Rectangle")
26 print("3. Triangle")
27
28 choice = int(input("Enter the number of your choice (1/2/3): "))
29
30 if choice == 1:
31     radius = float(input("Enter the radius of the circle: "))
32     area, perimeter = calculate_circle_area_and_perimeter(radius)
33     print(f"Area of the circle: (area:.2f)")
34     print(f"Perimeter of the circle: (perimeter:.2f)")
35
36 elif choice == 2:
37     length = float(input("Enter the length of the rectangle: "))
38     width = float(input("Enter the width of the rectangle: "))
39     area, perimeter = calculate_rectangle_area_and_perimeter(length, width)
40     print(f"Area of the rectangle: (area:.2f)")
41     print(f"Perimeter of the rectangle: (perimeter:.2f)")
42
43 elif choice == 3:
44     base = float(input("Enter the base of the triangle: "))
45     height = float(input("Enter the height of the triangle: "))
```

Second Screenshot: The code editor shows the same program, but with additional input prompts for the triangle's sides. The code is as follows:

```
10 def calculate_rectangle_area_and_perimeter(length, width):
11     area = length * width
12     perimeter = 2 * (length + width)
13     return area, perimeter
14
15 # Function to calculate the area and perimeter of a triangle
16 def calculate_triangle_area_and_perimeter(base, height, side1, side2, side3):
17     perimeter = side1 + side2 + side3
18     s = perimeter / 2 # Semi-perimeter for Heron's formula
19     area = math.sqrt(s * (s - side1) * (s - side2) * (s - side3))
20     return area, perimeter
21
22 # Menu for shape selection
23 print("Select a geometric shape:")
24 print("1. Circle")
25 print("2. Rectangle")
26 print("3. Triangle")
27
28 choice = int(input("Enter the number of your choice (1/2/3): "))
29
30 if choice == 1:
31     radius = float(input("Enter the radius of the circle: "))
32     area, perimeter = calculate_circle_area_and_perimeter(radius)
33     print(f"Area of the circle: (area:.2f)")
34     print(f"Perimeter of the circle: (perimeter:.2f)")
35
36 elif choice == 2:
37     length = float(input("Enter the length of the rectangle: "))
38     width = float(input("Enter the width of the rectangle: "))
39     area, perimeter = calculate_rectangle_area_and_perimeter(length, width)
40     print(f"Area of the rectangle: (area:.2f)")
41     print(f"Perimeter of the rectangle: (perimeter:.2f)")
42
43 elif choice == 3:
44     base = float(input("Enter the base of the triangle: "))
45     height = float(input("Enter the height of the triangle: "))
46     side1 = float(input("Enter the length of side 1: "))
47     side2 = float(input("Enter the length of side 2: "))
48     side3 = float(input("Enter the length of side 3: "))
49     area, perimeter = calculate_triangle_area_and_perimeter(base, height, side1, side2, side3)
50     print(f"Area of the triangle: (area:.2f)")
51     print(f"Perimeter of the triangle: (perimeter:.2f)")
52
53 else:
54     print("Invalid choice. Please select a valid option (1/2/3).")
```

8. Implement a python function to Convert Celsius to Fahrenheit and Fahrenheit to Celsius.



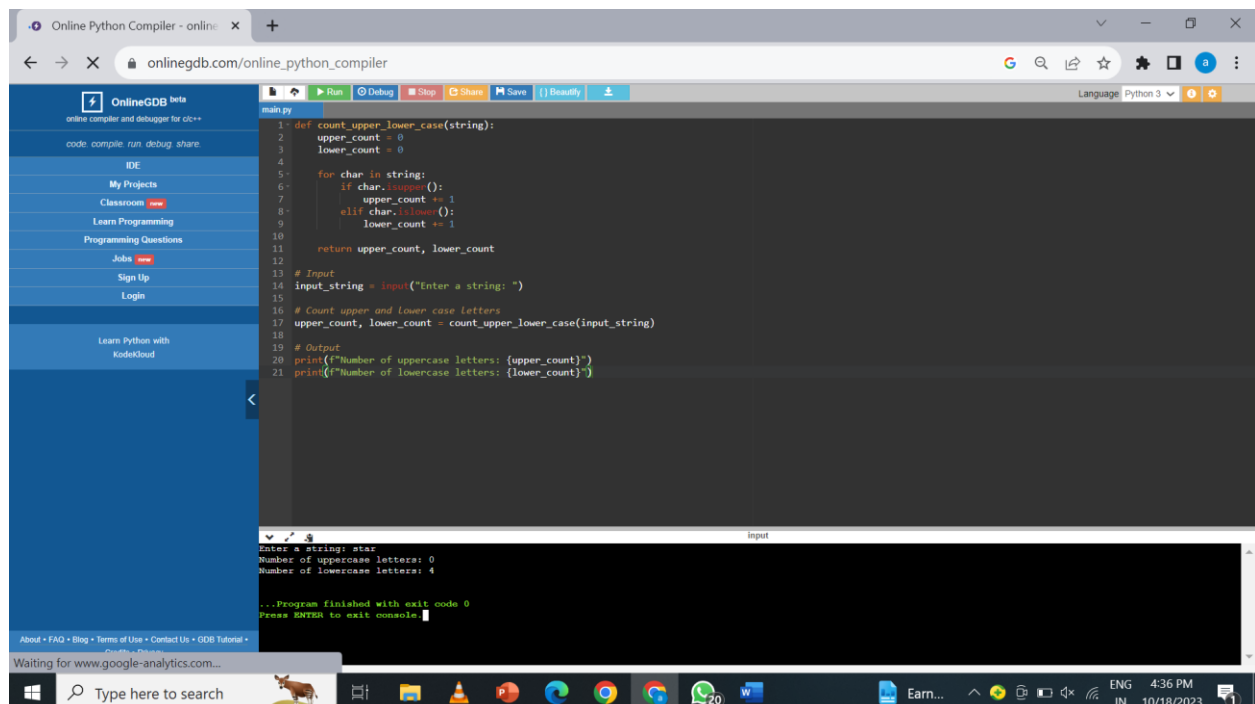
The screenshot shows the Online Python Compiler interface. The code defines two functions: `celsius_to_fahrenheit` and `fahrenheit_to_celsius`. It uses `input` to get the conversion type and temperature, then prints the result. The console output shows the user selecting Celsius to Fahrenheit and entering 34, resulting in 93.2.

```
main.py
1 # Function to convert Celsius to Fahrenheit
2 def celsius_to_fahrenheit(celsius):
3     fahrenheit = (celsius * 9/5) + 32
4     return fahrenheit
5
6 # Function to convert Fahrenheit to Celsius
7 def fahrenheit_to_celsius(fahrenheit):
8     celsius = (fahrenheit - 32) * 5/9
9     return celsius
10
11 # Input
12 choice = int(input("Select the conversion type:\n1. Celsius to Fahrenheit\n2. Fahrenheit to Celsius\nEnter 1 or 2: "))
13
14 if choice == 1:
15     celsius = float(input("Enter the temperature in Celsius: "))
16     fahrenheit = celsius_to_fahrenheit(celsius)
17     print(f"{celsius}°C is equal to {fahrenheit:.2f}°F.")
18 elif choice == 2:
19     fahrenheit = float(input("Enter the temperature in Fahrenheit: "))
20     celsius = fahrenheit_to_celsius(fahrenheit)
21     print(f"{fahrenheit}°F is equal to {celsius:.2f}°C.")
22 else:
23     print("Invalid choice. Please enter 1 or 2 for the conversion type.")
```

Input

```
Select the conversion type:
1. Celsius to Fahrenheit
2. Fahrenheit to Celsius
Enter 1 or 2: 1
Enter the temperature in Celsius: 34
93.2
```

9. Write a Python program that accepts a string and counts the number of upper and lower letters



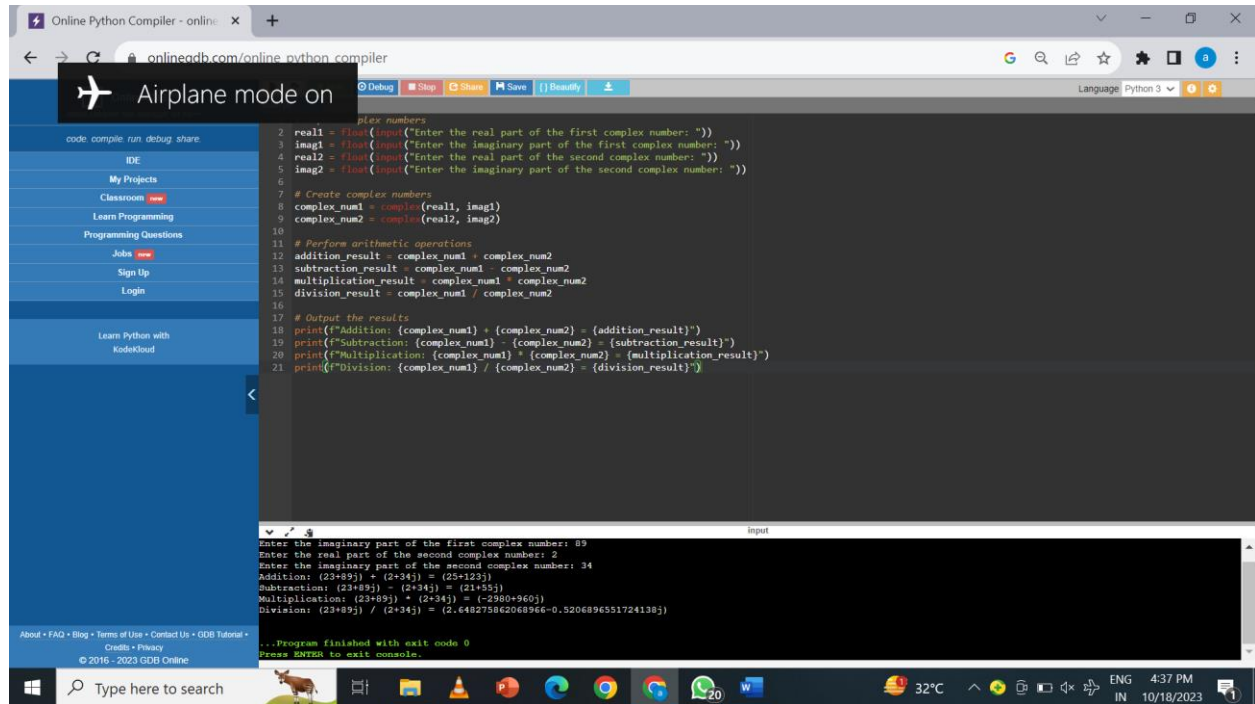
The screenshot shows the Online Python Compiler interface. The code defines a function `count_upper_lower_case` that counts the number of uppercase and lowercase letters in a string. It uses `input` to get the string and prints the counts. The console output shows the user entering 'star', resulting in 0 uppercase and 4 lowercase letters.

```
main.py
1 def count_upper_lower_case(string):
2     upper_count = 0
3     lower_count = 0
4
5     for char in string:
6         if char.isupper():
7             upper_count += 1
8         elif char.islower():
9             lower_count += 1
10
11     return upper_count, lower_count
12
13 # Input
14 input_string = input("Enter a string: ")
15
16 # Count upper and lower case letters
17 upper_count, lower_count = count_upper_lower_case(input_string)
18
19 # Output
20 print(f"Number of uppercase letters: {upper_count}")
21 print(f"Number of lowercase letters: {lower_count}")
```

Input

```
Enter a string: star
Number of uppercase letters: 0
Number of lowercase letters: 4
...Program finished with exit code 0
Press ENTER to exit console.
```

10. Write a python program to perform Arithmetic operations on Complex Numbers.



The screenshot shows a web browser window with the URL `onlinegdb.com/online_python_compiler`. The browser's address bar and tabs are visible at the top. The main content area displays a Python script for performing arithmetic operations on complex numbers. The script prompts the user to enter the real and imaginary parts of two complex numbers, calculates their sum, difference, product, and quotient, and prints the results. The output of the program is shown in a console window at the bottom, indicating that the program finished successfully with exit code 0.

```
1 # Python program to perform arithmetic operations on complex numbers
2 real1 = float(input("Enter the real part of the first complex number: "))
3 imag1 = float(input("Enter the imaginary part of the first complex number: "))
4 real2 = float(input("Enter the real part of the second complex number: "))
5 imag2 = float(input("Enter the imaginary part of the second complex number: "))
6
7 # Create complex numbers
8 complex_num1 = complex(real1, imag1)
9 complex_num2 = complex(real2, imag2)
10
11 # Perform arithmetic operations
12 addition_result = complex_num1 + complex_num2
13 subtraction_result = complex_num1 - complex_num2
14 multiplication_result = complex_num1 * complex_num2
15 division_result = complex_num1 / complex_num2
16
17 # Output the results
18 print(f"Addition: {complex_num1} + {complex_num2} = {addition_result}")
19 print(f"Subtraction: {complex_num1} - {complex_num2} = {subtraction_result}")
20 print(f"Multiplication: {complex_num1} * {complex_num2} = {multiplication_result}")
21 print(f"Division: {complex_num1} / {complex_num2} = {division_result}")
```

Enter the imaginary part of the first complex number: 89
Enter the real part of the second complex number: 2
Enter the imaginary part of the second complex number: 34
Addition: (23+89j) + (2+34j) = (25+123j)
Subtraction: (23+89j) - (2+34j) = (21+55j)
Multiplication: (23+89j) * (2+34j) = (-2980+960j)
Division: (23+89j) / (2+34j) = (2.648275862068966-0.5206896551724138j)

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