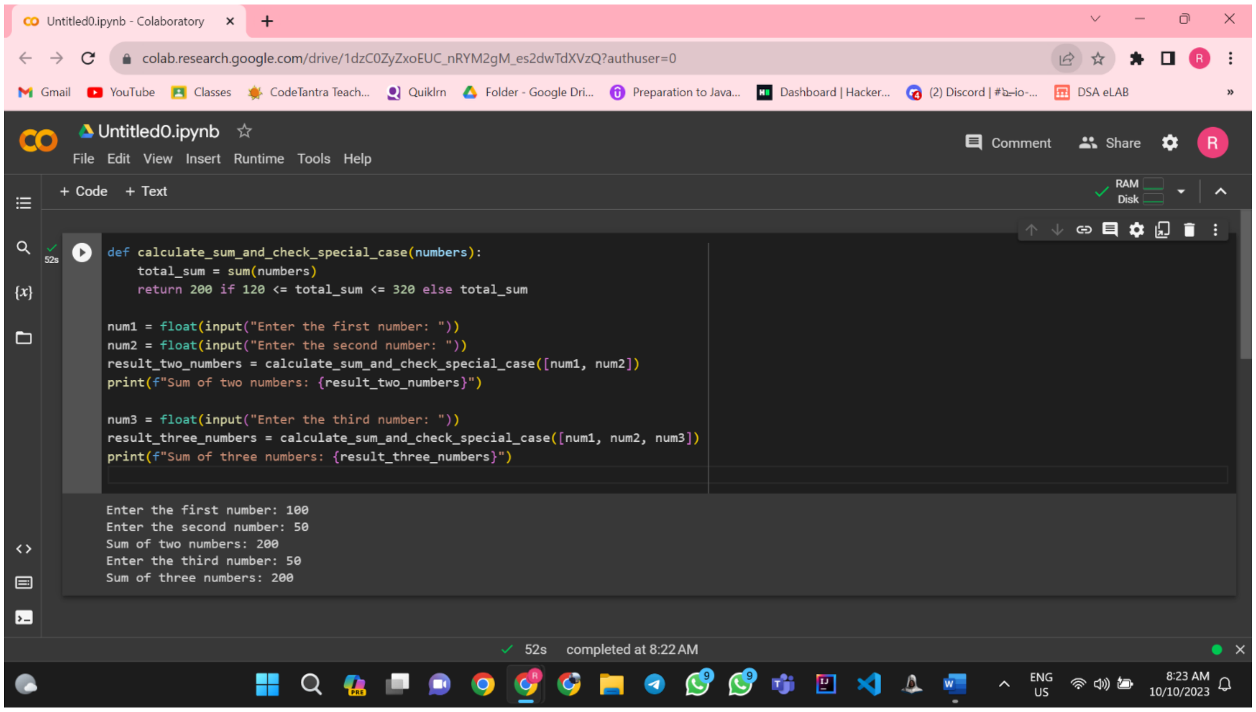
Rohan Soni

RA2211003012027

Week 10

1. Write a python program to calculate the sum of Two numbers and Three numbers.

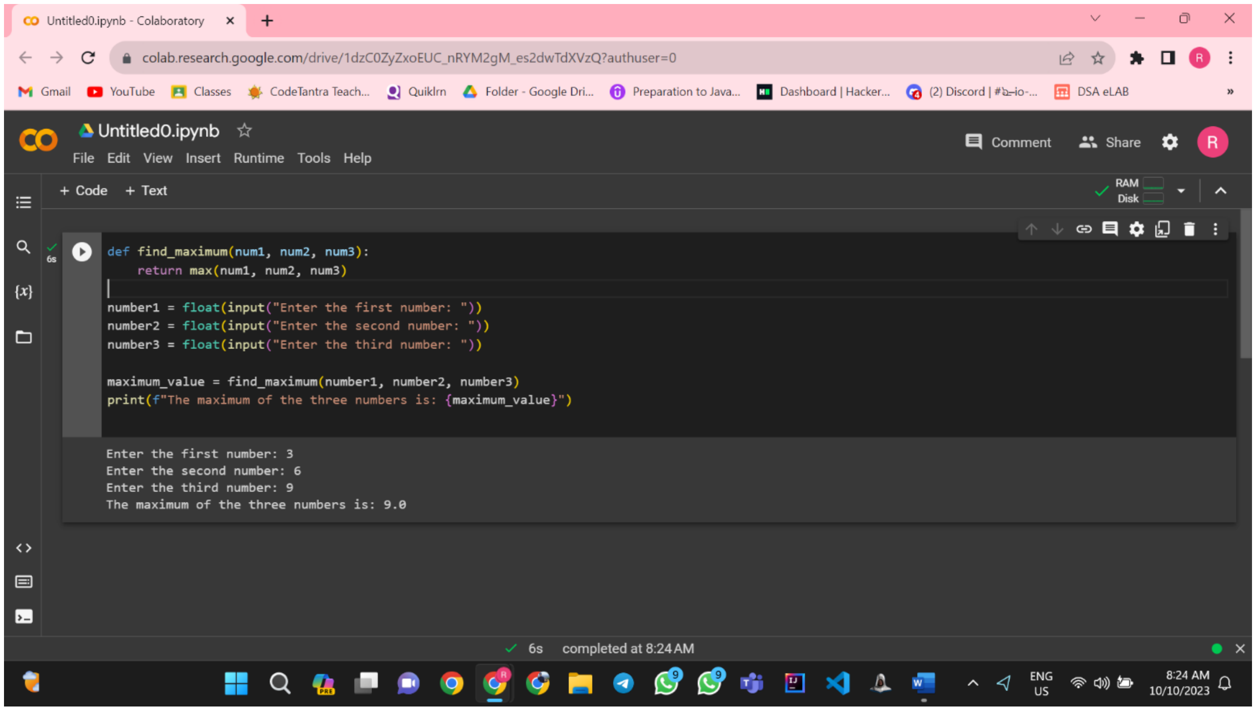
However, if the sum is between 120 to 320 it will return 200.



```python
def calculate_sum_and_check_special_case(numbers):
    total_sum = sum(numbers)
    return 200 if 120 <= total_sum <= 320 else total_sum

num1 = float(input("Enter the first number: "))
num2 = float(input("Enter the second number: "))
result_two_numbers = calculate_sum_and_check_special_case([num1, num2])
print(f"Sum of two numbers: {result_two_numbers}")

num3 = float(input("Enter the third number: "))
result_three_numbers = calculate_sum_and_check_special_case([num1, num2, num3])
print(f"Sum of three numbers: {result_three_numbers}")
```

```
Enter the first number: 100
Enter the second number: 50
Sum of two numbers: 200
Enter the third number: 50
Sum of three numbers: 200
```

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

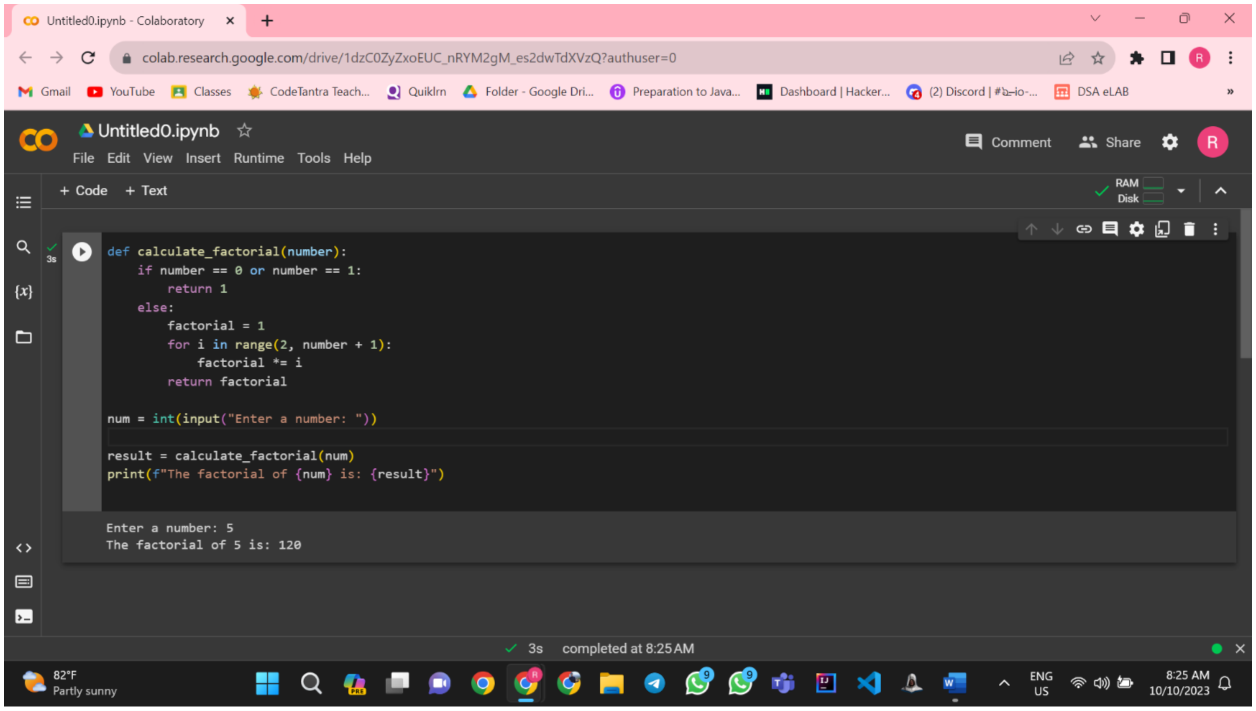


```python
def find_maximum(num1, num2, num3):
    return max(num1, num2, num3)

number1 = float(input("Enter the first number: "))
number2 = float(input("Enter the second number: "))
number3 = float(input("Enter the third number: "))

maximum_value = find_maximum(number1, number2, number3)
print(f"The maximum of the three numbers is: {maximum_value}")
```

```
Enter the first number: 3
Enter the second number: 6
Enter the third number: 9
The maximum of the three numbers is: 9.0
```

Untitled0.ipynb - Colaboratory
colab.research.google.com/drive/1dzC0ZyZxoEUC_nRYM2gM_es2dwTdXVzQ?authuser=0
Untitled0.ipynb
File Edit View Insert Runtime Tools Help
Comment Share
+ Code + Text
RAM Disk
6s completed at 8:24 AM
8:24 AM 10/10/2023

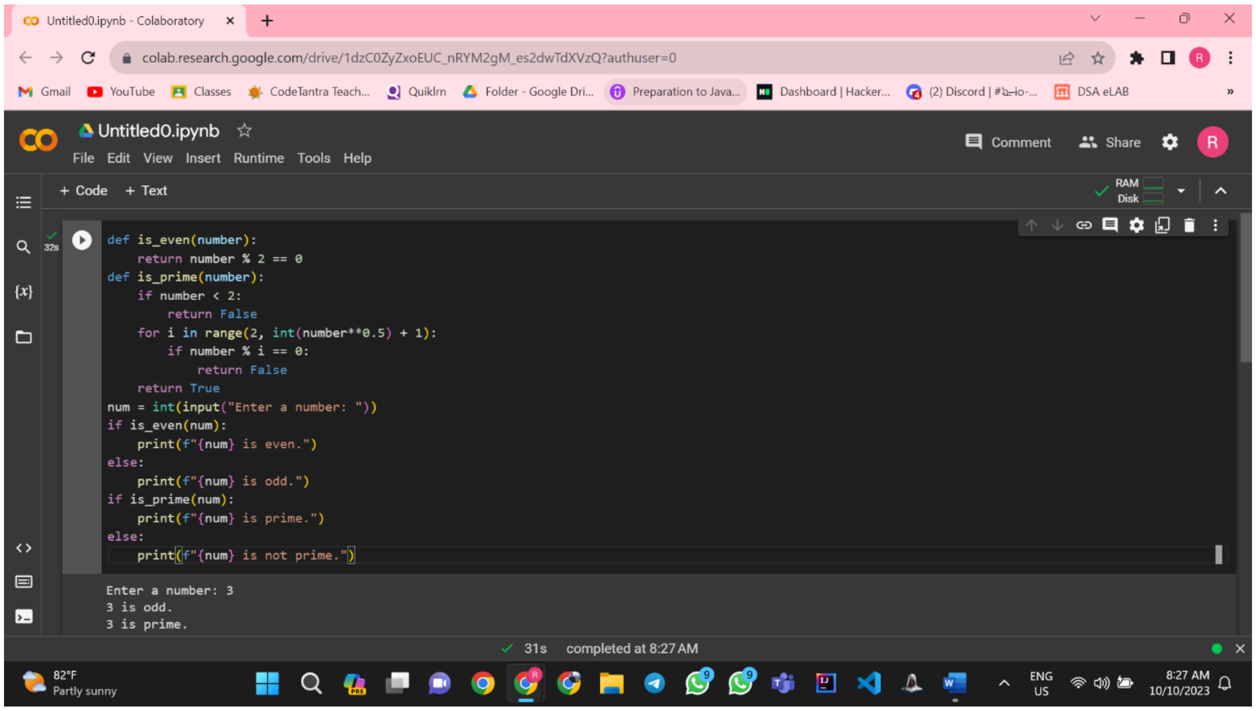
3. Write a python program to calculate the Factorial of a given number.



```
def calculate_factorial(number):
    if number == 0 or number == 1:
        return 1
    else:
        factorial = 1
        for i in range(2, number + 1):
            factorial *= i
        return factorial

num = int(input("Enter a number: "))

result = calculate_factorial(num)
print(f"The factorial of {num} is: {result}")
```

```
Enter a number: 5
The factorial of 5 is: 120
```

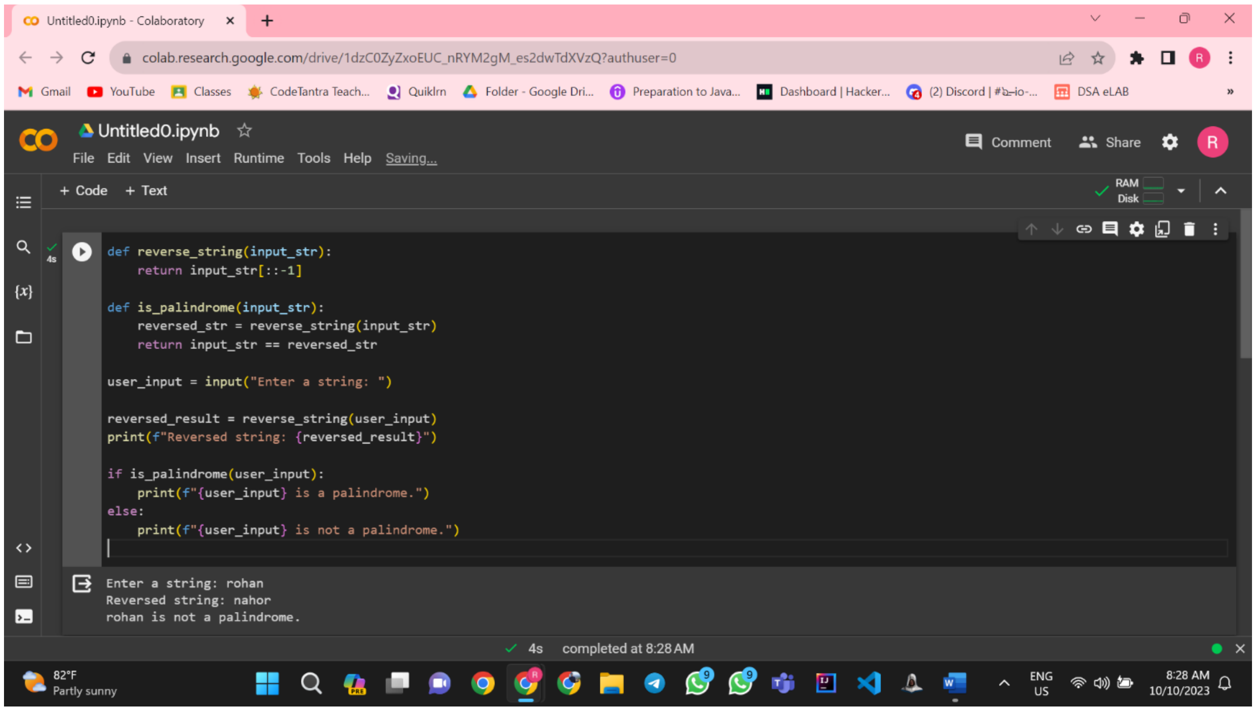
4. Write a python program to Check if a Number is Even or Odd and also check whether it is Prime or not.



```
def is_even(number):
    return number % 2 == 0
def is_prime(number):
    if number < 2:
        return False
    for i in range(2, int(number**0.5) + 1):
        if number % i == 0:
            return False
    return True
num = int(input("Enter a number: "))
if is_even(num):
    print(f"{num} is even.")
else:
    print(f"{num} is odd.")
if is_prime(num):
    print(f"{num} is prime.")
else:
    print(f"{num} is not prime.")
```

```
Enter a number: 3
3 is odd.
3 is prime.
```

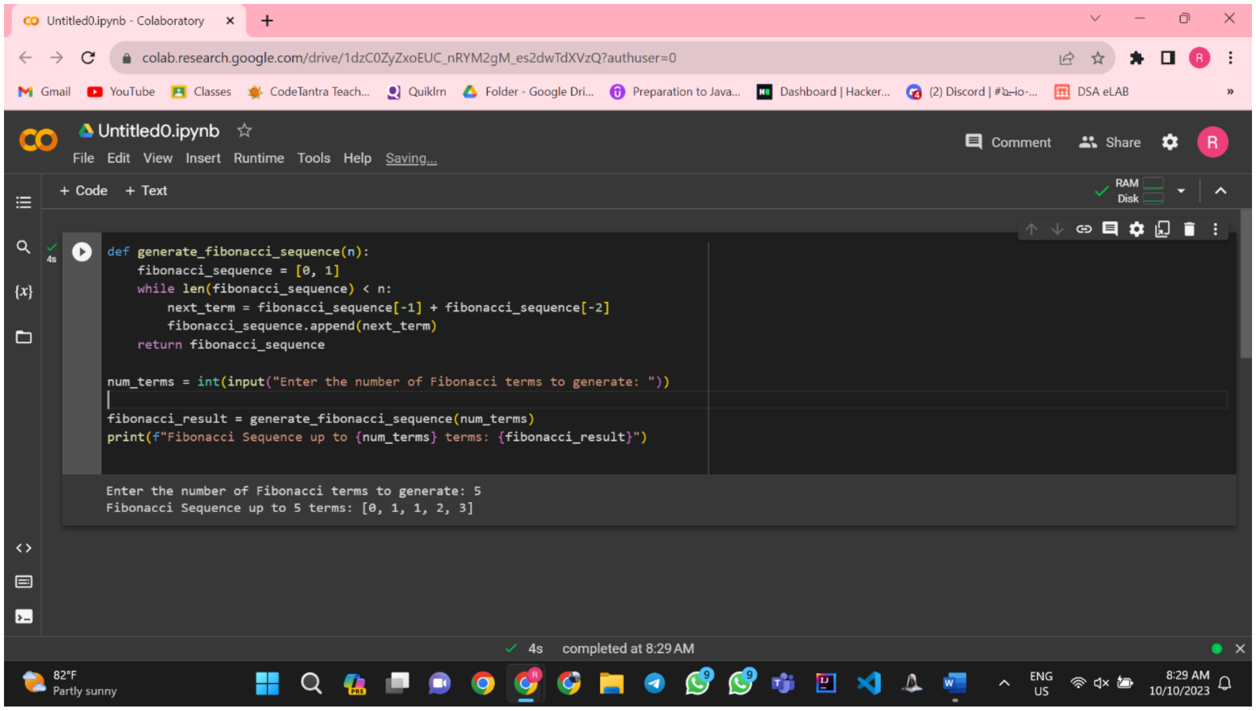
5. Implement a python function to Reverse a given String and also check for palindrome or not.


Untitled0.ipynb - Colaboratory
colab.research.google.com/drive/1dzC0ZyZxoEUC_nRYM2gM_es2dwTdXVzQ?authuser=0
Untitled0.ipynb
File Edit View Insert Runtime Tools Help Saving...
Comment Share
+ Code + Text
RAM Disk
4s completed at 8:28 AM
82°F Partly sunny
8:28 AM 10/10/2023

```python
def reverse_string(input_str):
    return input_str[::-1]

def is_palindrome(input_str):
    reversed_str = reverse_string(input_str)
    return input_str == reversed_str

user_input = input("Enter a string: ")

reversed_result = reverse_string(user_input)
print(f"Reversed string: {reversed_result}")

if is_palindrome(user_input):
    print(f"{user_input} is a palindrome.")
else:
    print(f"{user_input} is not a palindrome.")
```

```
Enter a string: rohan
Reversed string: nahor
rohan is not a palindrome.
```

6. Write a python program to Generate Fibonacci Sequence.

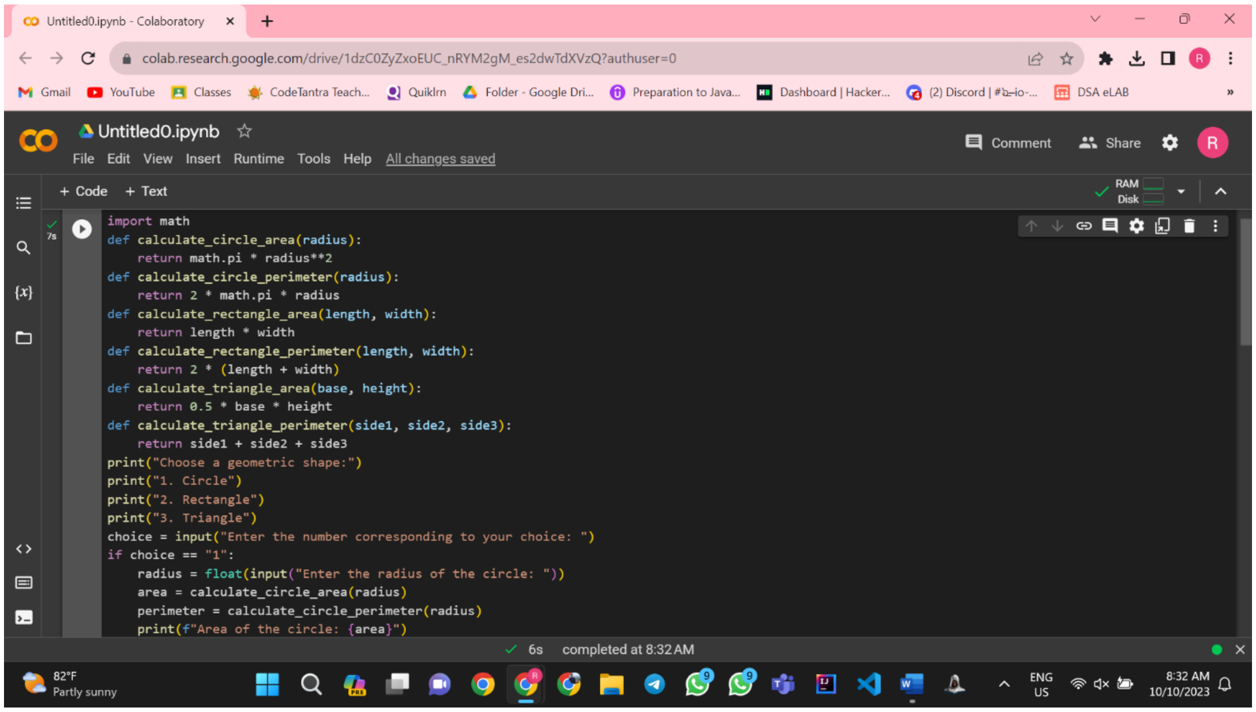


```python
def generate_fibonacci_sequence(n):
    fibonacci_sequence = [0, 1]
    while len(fibonacci_sequence) < n:
        next_term = fibonacci_sequence[-1] + fibonacci_sequence[-2]
        fibonacci_sequence.append(next_term)
    return fibonacci_sequence

num_terms = int(input("Enter the number of Fibonacci terms to generate: "))

fibonacci_result = generate_fibonacci_sequence(num_terms)
print(f"Fibonacci Sequence up to {num_terms} terms: {fibonacci_result}")
```

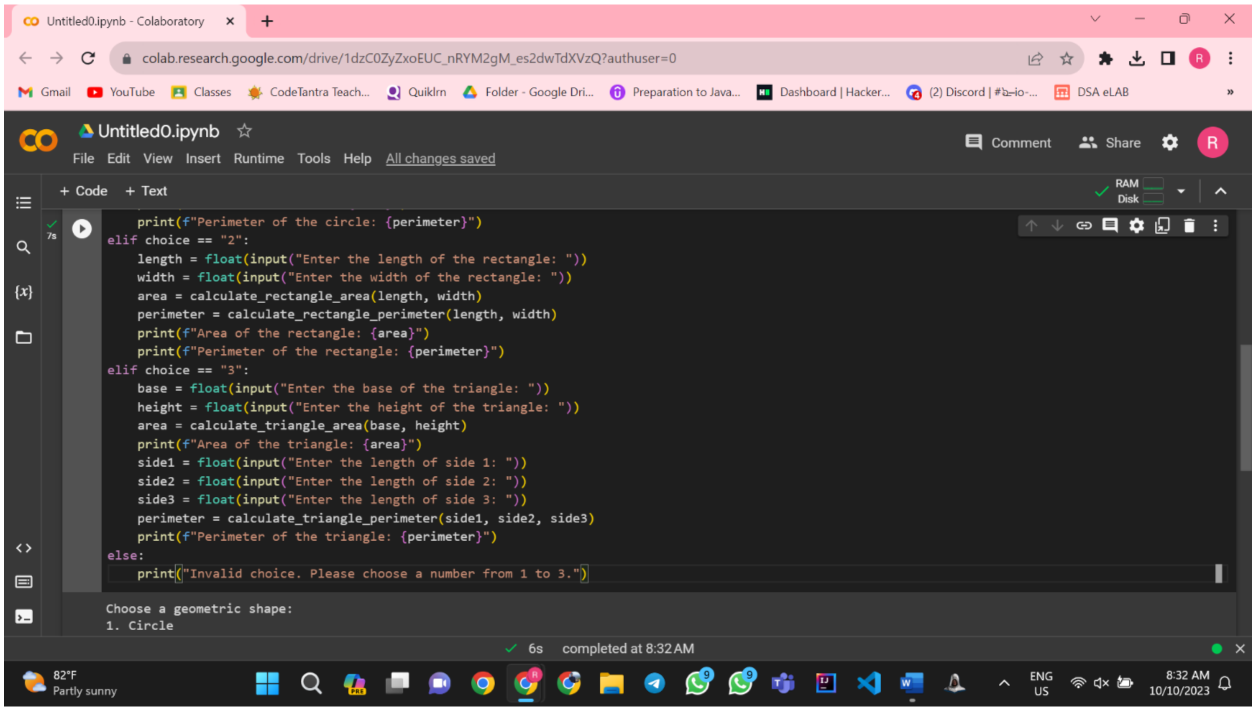
```
Enter the number of Fibonacci terms to generate: 5
Fibonacci Sequence up to 5 terms: [0, 1, 1, 2, 3]
```

7. Write a python program to calculate the area and perimeter of different geometric

shapes (circle, rectangle, triangle, etc.).

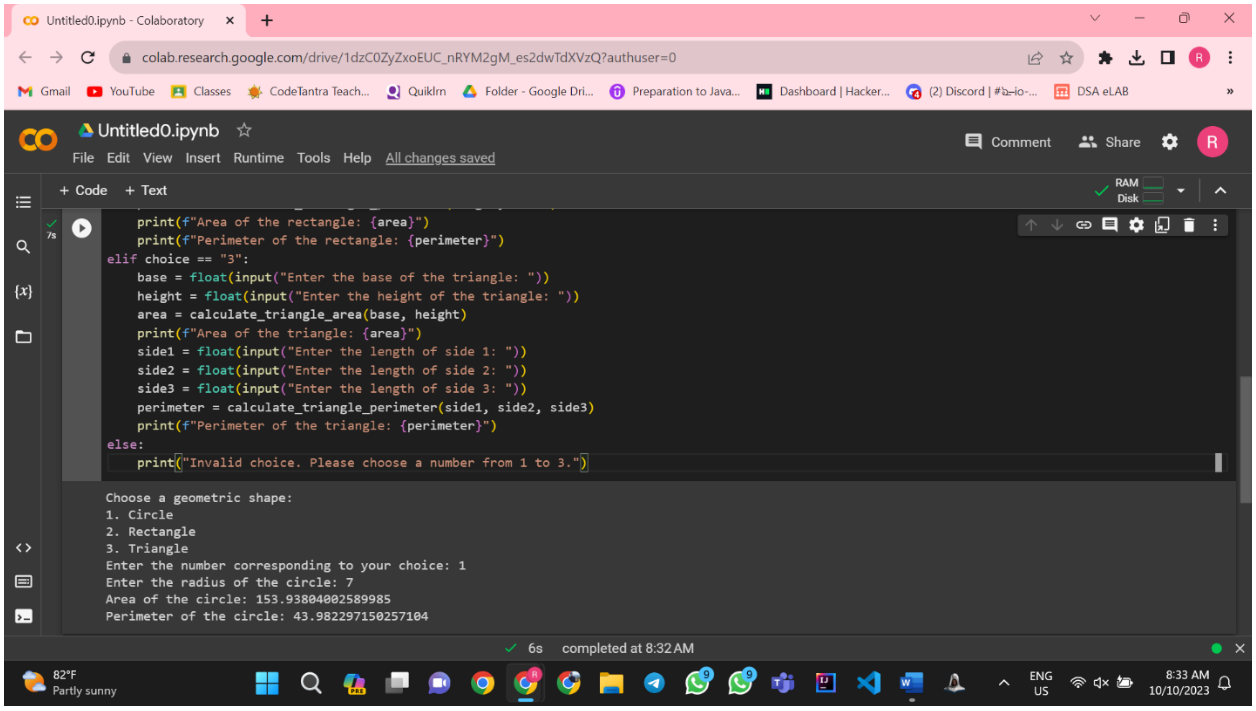

Untitled0.ipynb - Colaboratory
colab.research.google.com/drive/1dzC0ZyZxoEUC_nRYM2gM_es2dwTdXVzQ?authuser=0
Gmail YouTube Classes CodeTantra Teach... Quiklrn Folder - Google Dri... Preparation to Java... Dashboard | Hacker... (2) Discord DSA eLAB
Untitled0.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
RAM Disk
6s completed at 8:32 AM
82°F Partly sunny
ENG US
8:32 AM 10/10/2023

```python
import math
def calculate_circle_area(radius):
    return math.pi * radius**2
def calculate_circle_perimeter(radius):
    return 2 * math.pi * radius
def calculate_rectangle_area(length, width):
    return length * width
def calculate_rectangle_perimeter(length, width):
    return 2 * (length + width)
def calculate_triangle_area(base, height):
    return 0.5 * base * height
def calculate_triangle_perimeter(side1, side2, side3):
    return side1 + side2 + side3
print("Choose a geometric shape:")
print("1. Circle")
print("2. Rectangle")
print("3. Triangle")
choice = input("Enter the number corresponding to your choice: ")
if choice == "1":
    radius = float(input("Enter the radius of the circle: "))
    area = calculate_circle_area(radius)
    perimeter = calculate_circle_perimeter(radius)
    print(f"Area of the circle: {area}")
```



```
    print(f"Perimeter of the circle: {perimeter}")
elif choice == "2":
    length = float(input("Enter the length of the rectangle: "))
    width = float(input("Enter the width of the rectangle: "))
    area = calculate_rectangle_area(length, width)
    perimeter = calculate_rectangle_perimeter(length, width)
    print(f"Area of the rectangle: {area}")
    print(f"Perimeter of the rectangle: {perimeter}")
elif choice == "3":
    base = float(input("Enter the base of the triangle: "))
    height = float(input("Enter the height of the triangle: "))
    area = calculate_triangle_area(base, height)
    print(f"Area of the triangle: {area}")
    side1 = float(input("Enter the length of side 1: "))
    side2 = float(input("Enter the length of side 2: "))
    side3 = float(input("Enter the length of side 3: "))
    perimeter = calculate_triangle_perimeter(side1, side2, side3)
    print(f"Perimeter of the triangle: {perimeter}")
else:
    print("Invalid choice. Please choose a number from 1 to 3.")
```

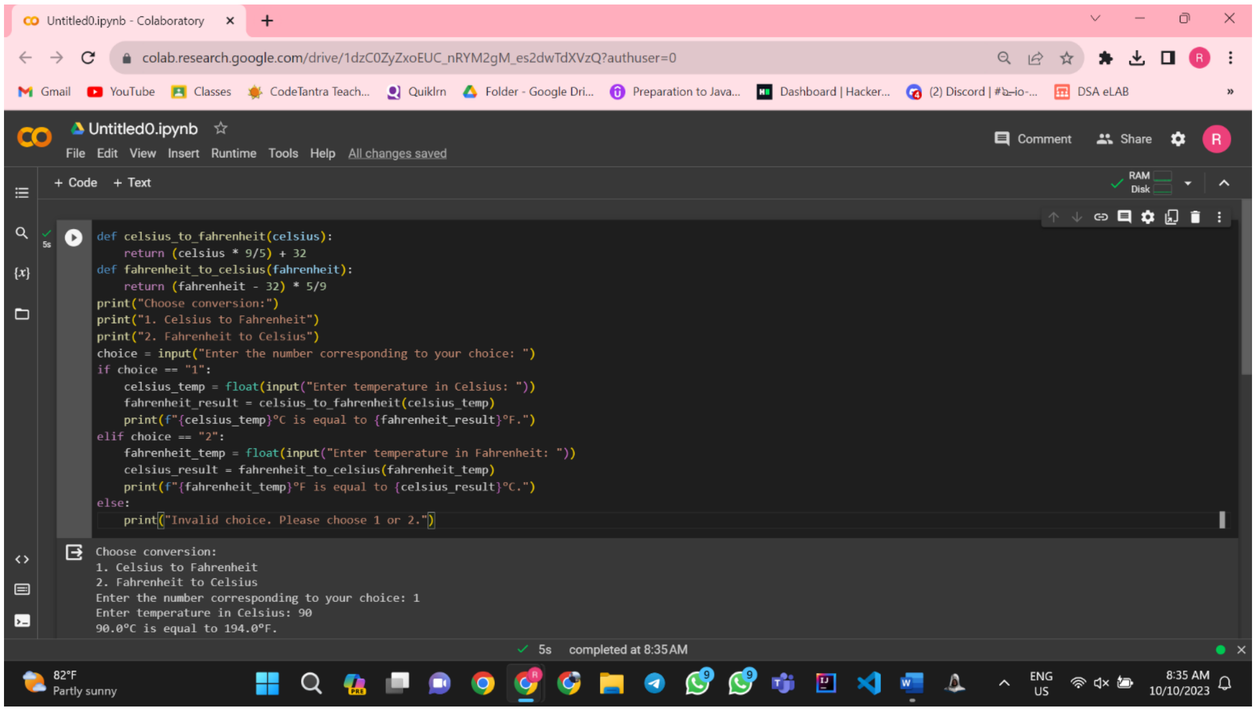
```
Choose a geometric shape:
1. Circle
```



```
    print(f"Area of the rectangle: {area}")
    print(f"Perimeter of the rectangle: {perimeter}")
elif choice == "3":
    base = float(input("Enter the base of the triangle: "))
    height = float(input("Enter the height of the triangle: "))
    area = calculate_triangle_area(base, height)
    print(f"Area of the triangle: {area}")
    side1 = float(input("Enter the length of side 1: "))
    side2 = float(input("Enter the length of side 2: "))
    side3 = float(input("Enter the length of side 3: "))
    perimeter = calculate_triangle_perimeter(side1, side2, side3)
    print(f"Perimeter of the triangle: {perimeter}")
else:
    print("Invalid choice. Please choose a number from 1 to 3.")
```

```
Choose a geometric shape:
1. Circle
2. Rectangle
3. Triangle
Enter the number corresponding to your choice: 1
Enter the radius of the circle: 7
Area of the circle: 153.93804002589985
Perimeter of the circle: 43.982297150257104
```

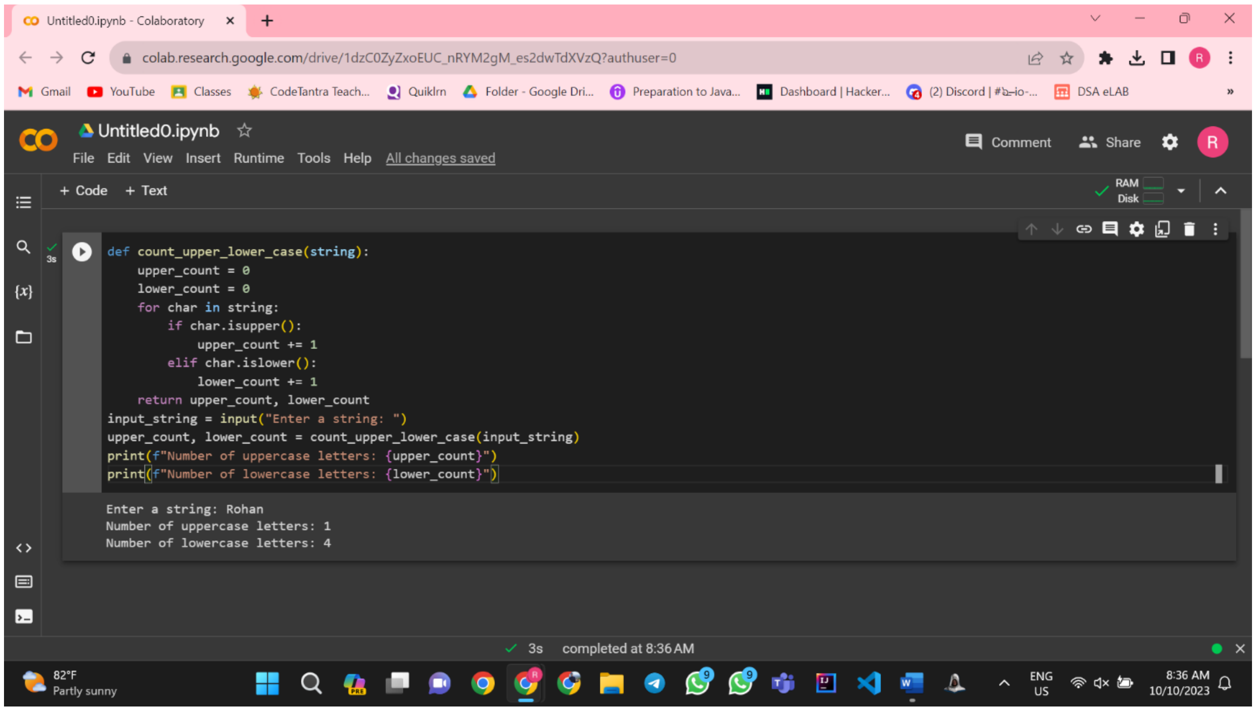
8. Implement a python function to Convert Celsius to Fahrenheit and Fahrenheit to Celsius.



```python
def celsius_to_fahrenheit(celsius):
    return (celsius * 9/5) + 32
def fahrenheit_to_celsius(fahrenheit):
    return (fahrenheit - 32) * 5/9
print("Choose conversion:")
print("1. Celsius to Fahrenheit")
print("2. Fahrenheit to Celsius")
choice = input("Enter the number corresponding to your choice: ")
if choice == "1":
    celsius_temp = float(input("Enter temperature in Celsius: "))
    fahrenheit_result = celsius_to_fahrenheit(celsius_temp)
    print(f"{celsius_temp}°C is equal to {fahrenheit_result}°F.")
elif choice == "2":
    fahrenheit_temp = float(input("Enter temperature in Fahrenheit: "))
    celsius_result = fahrenheit_to_celsius(fahrenheit_temp)
    print(f"{fahrenheit_temp}°F is equal to {celsius_result}°C.")
else:
    print("Invalid choice. Please choose 1 or 2.")
```

```
Choose conversion:
1. Celsius to Fahrenheit
2. Fahrenheit to Celsius
Enter the number corresponding to your choice: 1
Enter temperature in Celsius: 90
90.0°C is equal to 194.0°F.
```

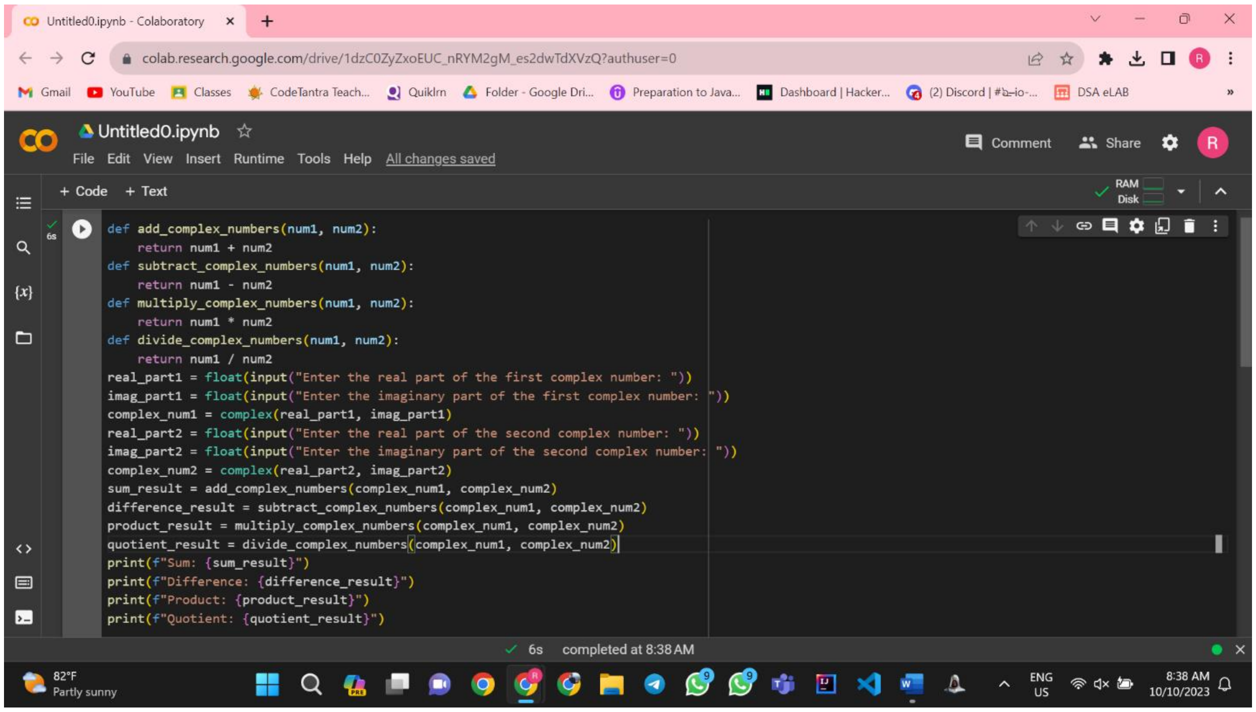
9. Write a Python program that accepts a string and counts the number of upper and lower case letters.



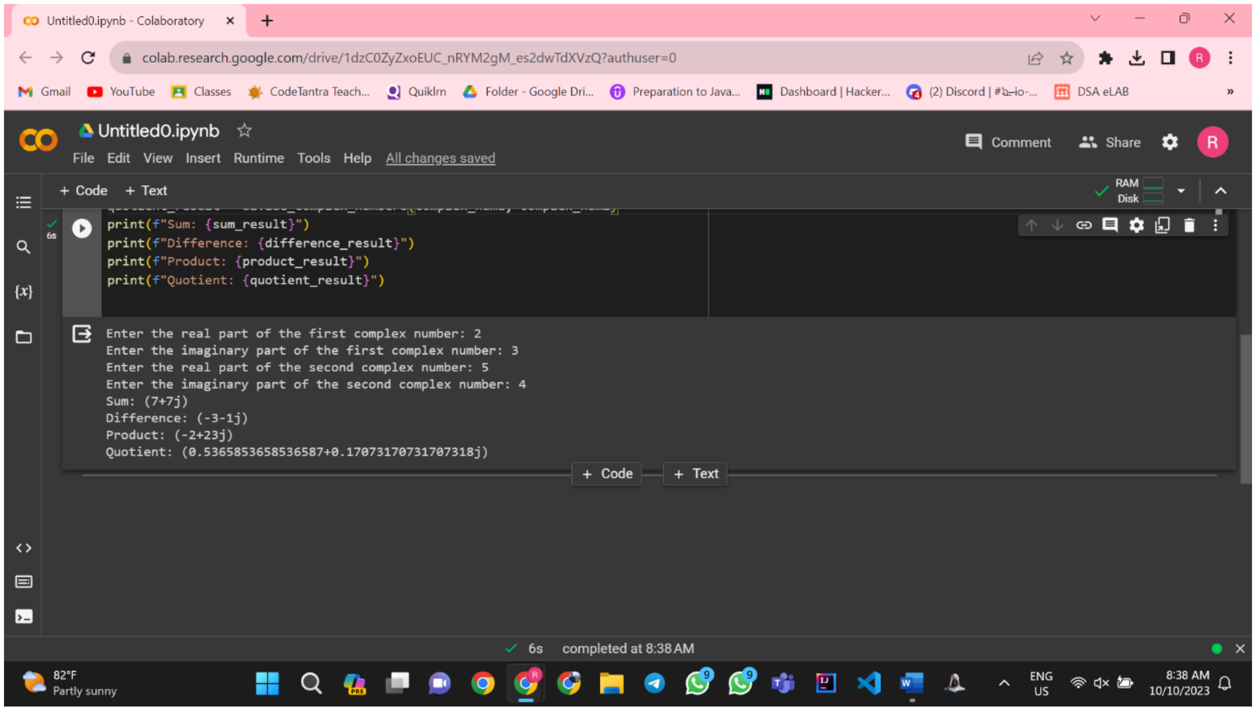
```python
def count_upper_lower_case(string):
    upper_count = 0
    lower_count = 0
    for char in string:
        if char.isupper():
            upper_count += 1
        elif char.islower():
            lower_count += 1
    return upper_count, lower_count
input_string = input("Enter a string: ")
upper_count, lower_count = count_upper_lower_case(input_string)
print(f"Number of uppercase letters: {upper_count}")
print(f"Number of lowercase letters: {lower_count}")
```

```
Enter a string: Rohan
Number of uppercase letters: 1
Number of lowercase letters: 4
```

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



```python
def add_complex_numbers(num1, num2):
    return num1 + num2
def subtract_complex_numbers(num1, num2):
    return num1 - num2
def multiply_complex_numbers(num1, num2):
    return num1 * num2
def divide_complex_numbers(num1, num2):
    return num1 / num2
real_part1 = float(input("Enter the real part of the first complex number: "))
imag_part1 = float(input("Enter the imaginary part of the first complex number: "))
complex_num1 = complex(real_part1, imag_part1)
real_part2 = float(input("Enter the real part of the second complex number: "))
imag_part2 = float(input("Enter the imaginary part of the second complex number: "))
complex_num2 = complex(real_part2, imag_part2)
sum_result = add_complex_numbers(complex_num1, complex_num2)
difference_result = subtract_complex_numbers(complex_num1, complex_num2)
product_result = multiply_complex_numbers(complex_num1, complex_num2)
quotient_result = divide_complex_numbers(complex_num1, complex_num2)
print(f"Sum: {sum_result}")
print(f"Difference: {difference_result}")
print(f"Product: {product_result}")
print(f"Quotient: {quotient_result}")
```

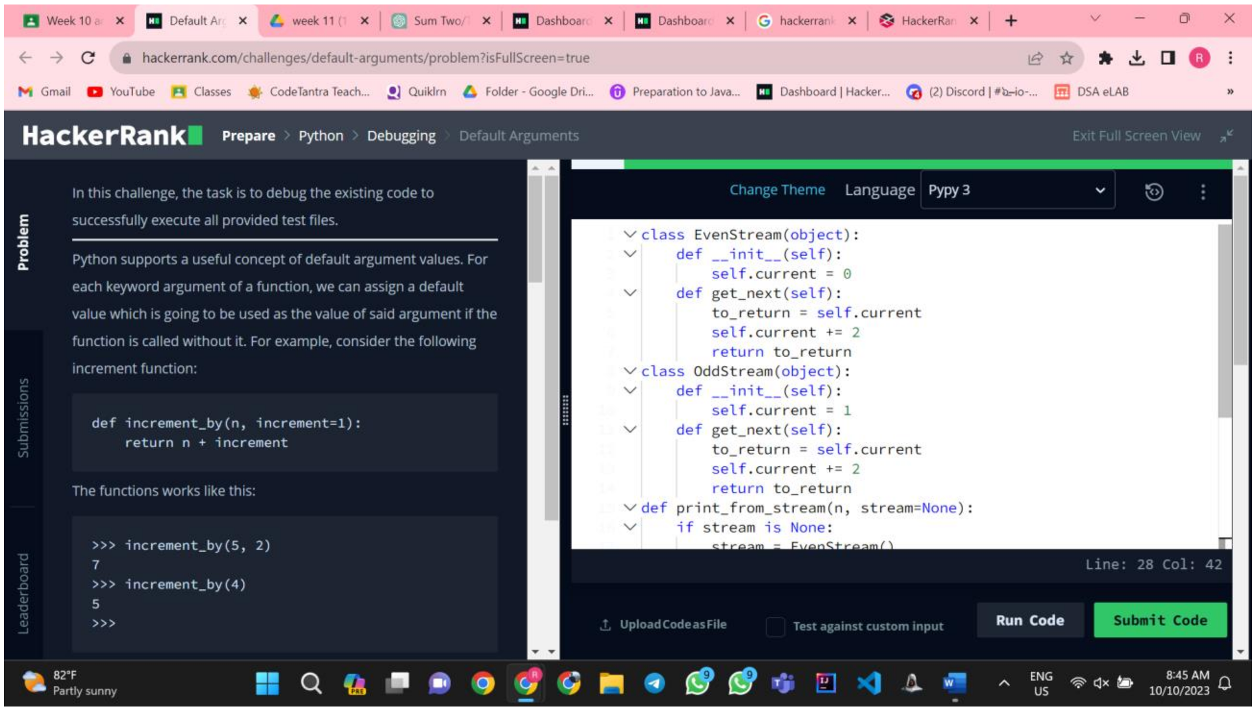


```python
print(f"Sum: {sum_result}")
print(f"Difference: {difference_result}")
print(f"Product: {product_result}")
print(f"Quotient: {quotient_result}")
```

```
Enter the real part of the first complex number: 2
Enter the imaginary part of the first complex number: 3
Enter the real part of the second complex number: 5
Enter the imaginary part of the second complex number: 4
Sum: (7+7j)
Difference: (-3-1j)
Product: (-2+23j)
Quotient: (0.5365853658536587+0.17073170731707318j)
```

HACKERRANK

https://www.hackerrank.com/challenges/default-arguments/problem?isFullScreen=true


hackerrank.com/challenges/default-arguments/problem?isFullScreen=true
HackerRank
Prepare > Python > Debugging > Default Arguments
Exit Full Screen View
Problem
Submissions
Leaderboard
Change Theme
Language
Pypy 3
Upload Code as File
Test against custom input
Run Code
Submit Code

In this challenge, the task is to debug the existing code to successfully execute all provided test files.

Python supports a useful concept of default argument values. For each keyword argument of a function, we can assign a default value which is going to be used as the value of said argument if the function is called without it. For example, consider the following increment function:

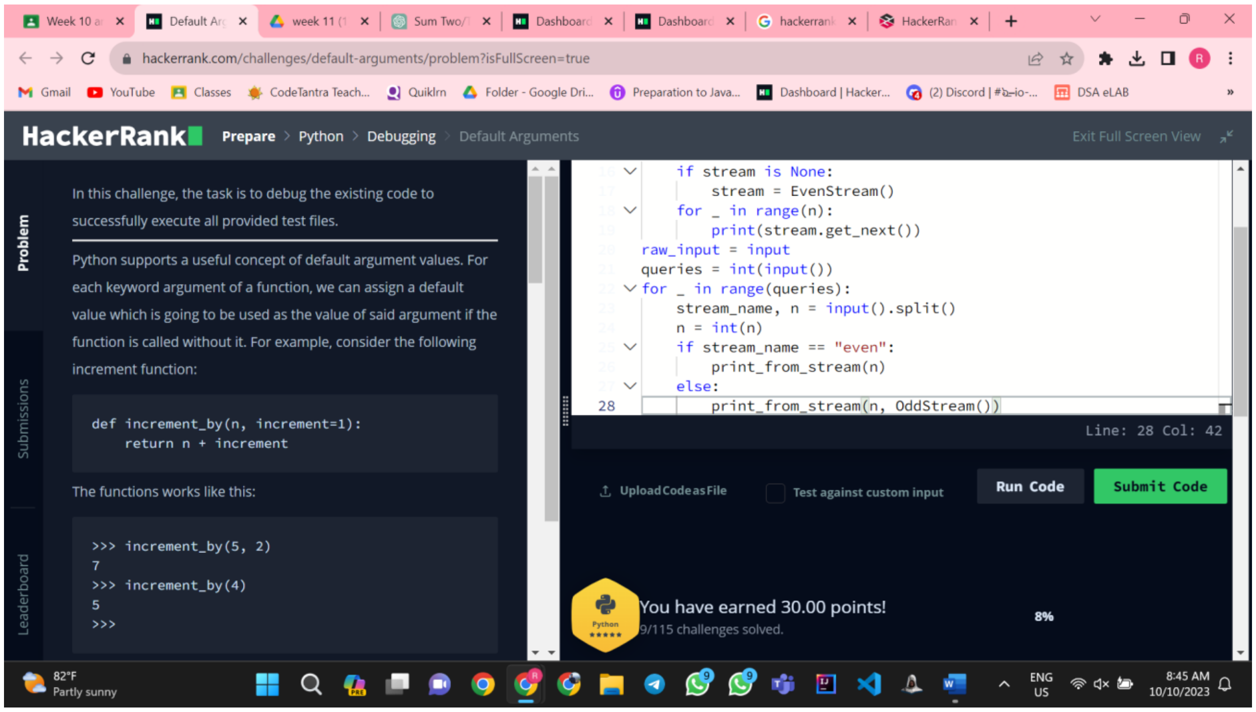
```
def increment_by(n, increment=1):
    return n + increment
```

The functions works like this:

```
>>> increment_by(5, 2)
7
>>> increment_by(4)
5
>>>
```

```
class EvenStream(object):
    def __init__(self):
        self.current = 0
    def get_next(self):
        to_return = self.current
        self.current += 2
        return to_return
class OddStream(object):
    def __init__(self):
        self.current = 1
    def get_next(self):
        to_return = self.current
        self.current += 2
        return to_return
def print_from_stream(n, stream=None):
    if stream is None:
        stream = EvenStream()
```

Line: 28 Col: 42


hackerrank.com/challenges/default-arguments/problem?isFullScreen=true
HackerRank
Prepare > Python > Debugging > Default Arguments
Exit Full Screen View
Problem
Submissions
Leaderboard
Upload Code as File
Test against custom input
Run Code
Submit Code
Python

In this challenge, the task is to debug the existing code to successfully execute all provided test files.

Python supports a useful concept of default argument values. For each keyword argument of a function, we can assign a default value which is going to be used as the value of said argument if the function is called without it. For example, consider the following increment function:

```
def increment_by(n, increment=1):
    return n + increment
```

The functions works like this:

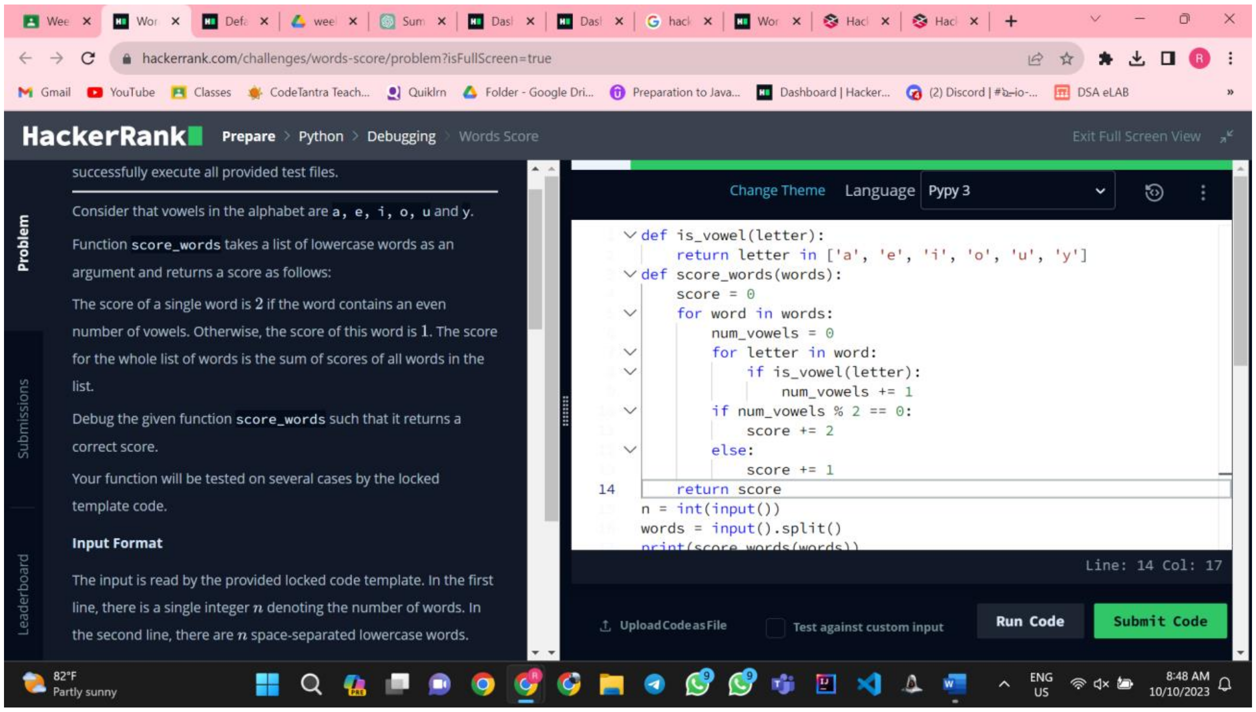
```
>>> increment_by(5, 2)
7
>>> increment_by(4)
5
>>>
```

```
    if stream is None:
        stream = EvenStream()
    for _ in range(n):
        print(stream.get_next())
raw_input = input
queries = int(input())
for _ in range(queries):
    stream_name, n = input().split()
    n = int(n)
    if stream_name == "even":
        print_from_stream(n)
    else:
        print_from_stream(n, OddStream())
```

Line: 28 Col: 42

You have earned 30.00 points!
9/115 challenges solved.
8%

https://www.hackerrank.com/challenges/words-score/problem?isFullScreen=true



HackerRank Prepare > Python > Debugging > Words Score

successfully execute all provided test files.

Consider that vowels in the alphabet are `a`, `e`, `i`, `o`, `u` and `y`.

Function `score_words` takes a list of lowercase words as an argument and returns a score as follows:

The score of a single word is 2 if the word contains an even number of vowels. Otherwise, the score of this word is 1. The score for the whole list of words is the sum of scores of all words in the list.

Debug the given function `score_words` such that it returns a correct score.

Your function will be tested on several cases by the locked template code.

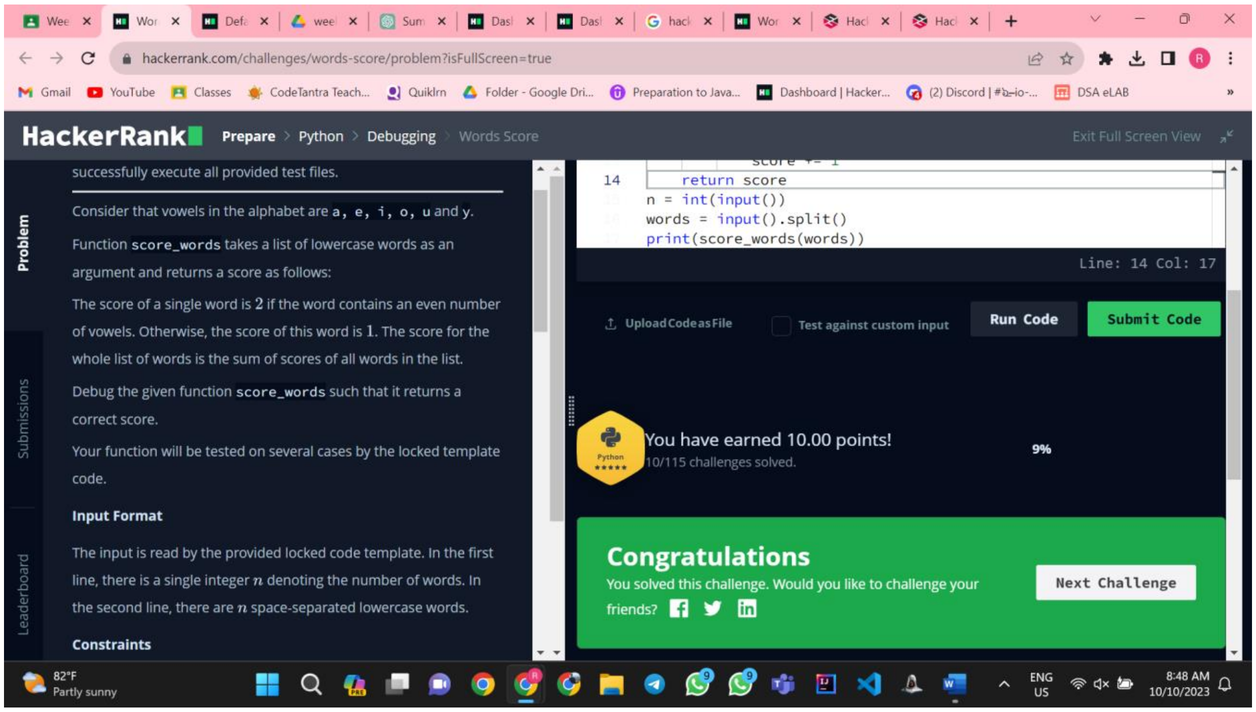
Input Format

The input is read by the provided locked code template. In the first line, there is a single integer n denoting the number of words. In the second line, there are n space-separated lowercase words.

Language Pypy 3

```
def is_vowel(letter):
    return letter in ['a', 'e', 'i', 'o', 'u', 'y']
def score_words(words):
    score = 0
    for word in words:
        num_vowels = 0
        for letter in word:
            if is_vowel(letter):
                num_vowels += 1
        if num_vowels % 2 == 0:
            score += 2
        else:
            score += 1
    return score
n = int(input())
words = input().split()
print(score_words(words))
```

Line: 14 Col: 17



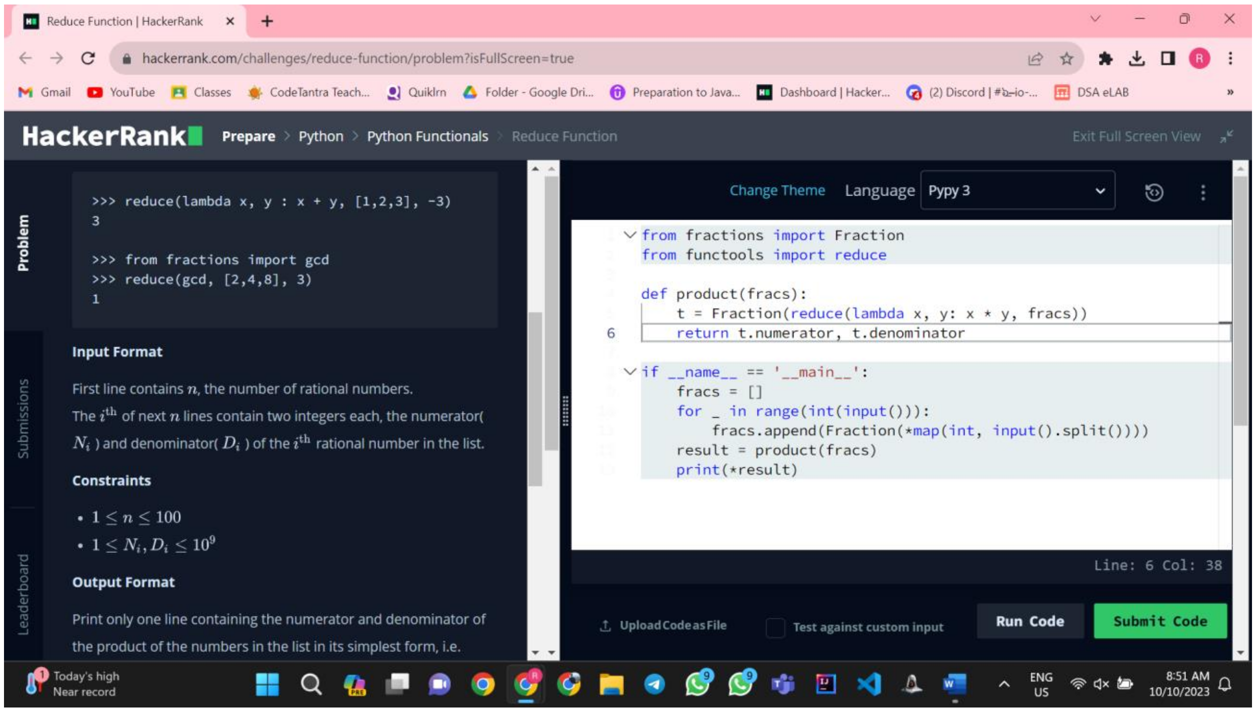
```
    return score
n = int(input())
words = input().split()
print(score_words(words))
```

You have earned 10.00 points!
10/115 challenges solved.

Congratulations

You solved this challenge. Would you like to challenge your friends?

https://www.hackerrank.com/challenges/reduce-function/problem?isFullScreen=true



```
>>> reduce(lambda x, y : x + y, [1,2,3], -3)
3

>>> from fractions import gcd
>>> reduce(gcd, [2,4,8], 3)
1
```

Input Format

First line contains n, the number of rational numbers.

The i^{th} of next n lines contain two integers each, the numerator(N_i) and denominator(D_i) of the i^{th} rational number in the list.

Constraints

- $1 \le n \le 100$
- $1 \le N_i, D_i \le 10^9$

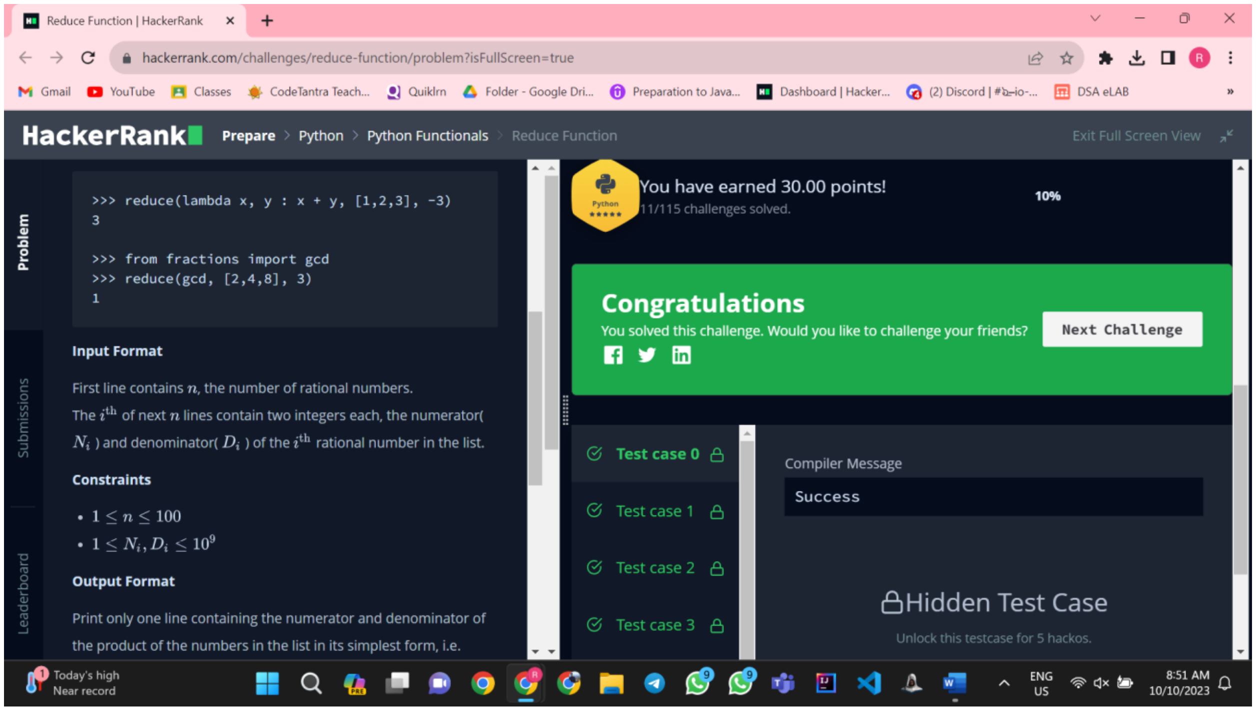
Output Format

Print only one line containing the numerator and denominator of the product of the numbers in the list in its simplest form, i.e.

Language: Pypy 3

```
from fractions import Fraction
from functools import reduce

def product(fracs):
    t = Fraction(reduce(lambda x, y: x * y, fracs))
    return t.numerator, t.denominator

if __name__ == '__main__':
    fracs = []
    for _ in range(int(input())):
        fracs.append(Fraction(*map(int, input().split())))
    result = product(fracs)
    print(*result)
```

Line: 6 Col: 38



You have earned 30.00 points!
11/115 challenges solved.

Congratulations

You solved this challenge. Would you like to challenge your friends?

Test case 0 ✓
Test case 1 ✓
Test case 2 ✓
Test case 3 ✓

Compiler Message

```
Success
```

Hidden Test Case

Unlock this testcase for 5 hackos.