AI Assisted Coding

Lab Assignment 4.2

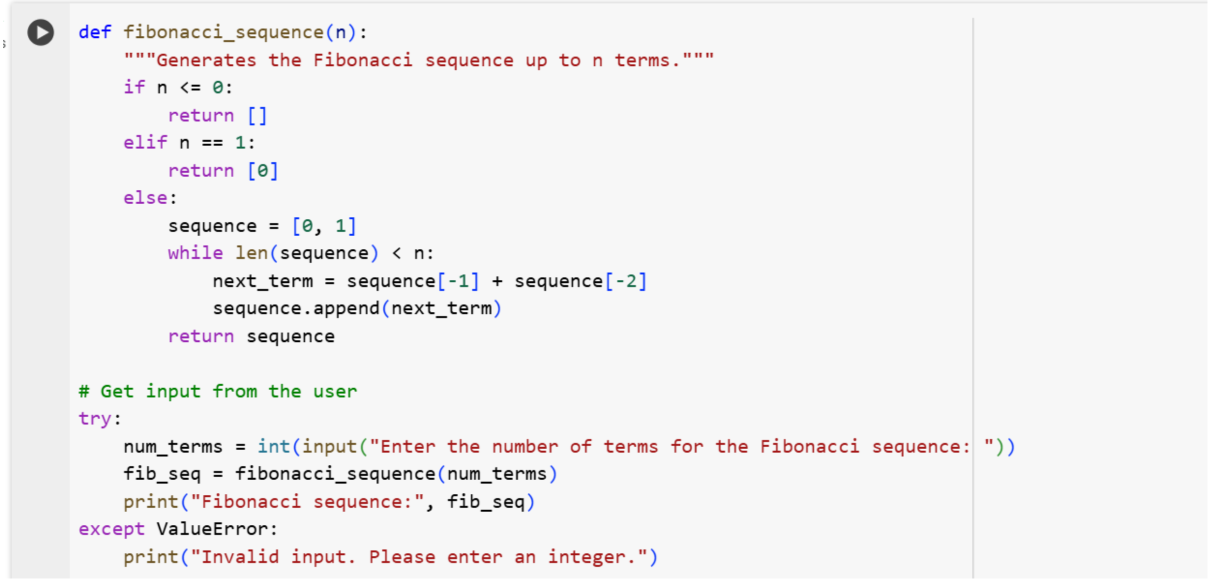
Name:A.Manideep

EN NO:2403a52078

Task 1: Zero-shot: Prompt AI with only the instruction — Write a Python function to generate the Fibonacci sequence up to n term.

PROMPT : GENERATE A PYTHON CODE FOR FIBONACCI SERIES SEQUENCE UP TO N TERM AND THE INPUT SHOULD TAKE FROM THE USER.

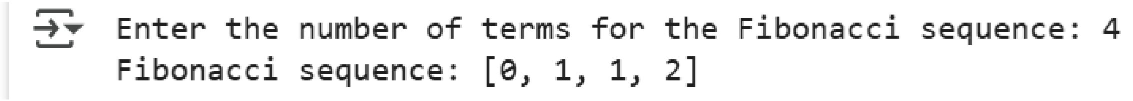
CODE GENERATED:



```python
def fibonacci_sequence(n):
    """Generates the Fibonacci sequence up to n terms."""
    if n <= 0:
        return []
    elif n == 1:
        return [0]
    else:
        sequence = [0, 1]
        while len(sequence) < n:
            next_term = sequence[-1] + sequence[-2]
            sequence.append(next_term)
        return sequence

# Get input from the user
try:
    num_terms = int(input("Enter the number of terms for the Fibonacci sequence: "))
    fib_seq = fibonacci_sequence(num_terms)
    print("Fibonacci sequence:", fib_seq)
except ValueError:
    print("Invalid input. Please enter an integer.")
```

OUTPUT:



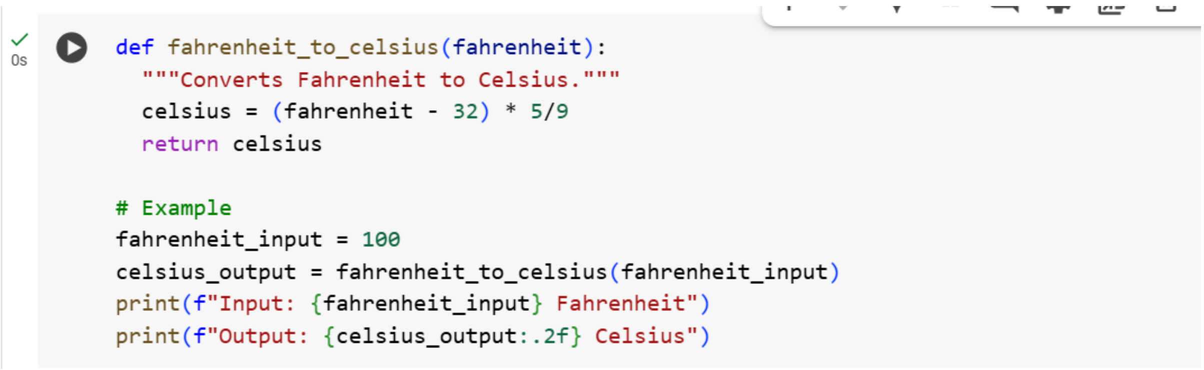
```
Enter the number of terms for the Fibonacci sequence: 4
Fibonacci sequence: [0, 1, 1, 2]
```

EXPLANATION: This code defines a function fibonacci_sequence that generates the Fibonacci sequence up to a specified number of terms n. It handles edge cases for n <= 0 and n == 1. For n > 1. The code then prompts the user for the number of terms, calls the function, and prints the resulting sequence.

Task 2: One-shot: Provide one example: Input: 100, Output: 37.78 to help AI generate a function that converts Fahrenheit to Celsius

PROMPT : GENERATE A PYTHON CODE FOR CONVERTING FAHRENHEIT TO CELSIUS PROVIDE ONE EXAMPLE: INPUT: 100, OUTPUT: 37.78

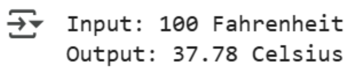
CODE GENERATED:



```python
def fahrenheit_to_celsius(fahrenheit):
  """Converts Fahrenheit to Celsius."""
  celsius = (fahrenheit - 32) * 5/9
  return celsius

# Example
fahrenheit_input = 100
celsius_output = fahrenheit_to_celsius(fahrenheit_input)
print(f"Input: {fahrenheit_input} Fahrenheit")
print(f"Output: {celsius_output:.2f} Celsius")
```

OUTPUT:



```
Input: 100 Fahrenheit
Output: 37.78 Celsius
```

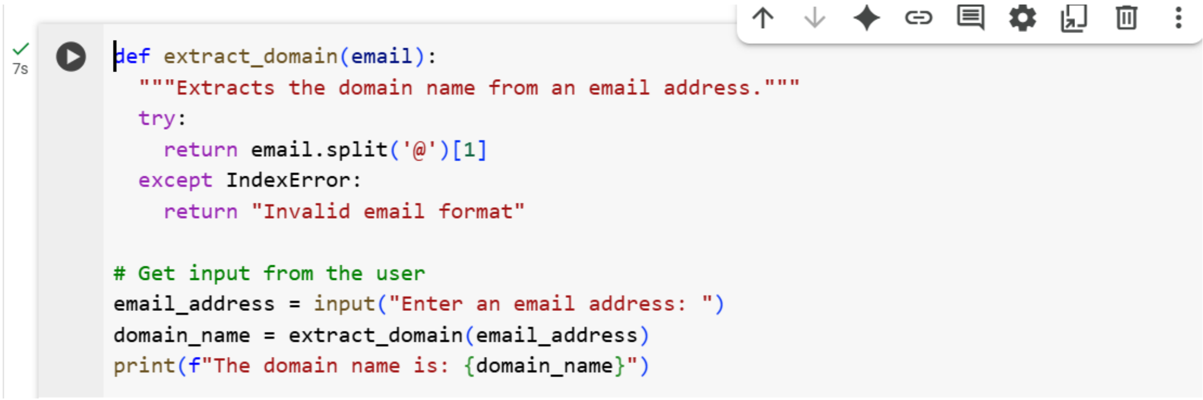
EXPLANATION:

THIS CODE DEFINES A FUNCTION FAHRENHEIT_TO_CELSIUS THAT TAKES A TEMPERATURE IN FAHRENHEIT AS INPUT APPLIES THE STANDARD FORMULA TO CONVERT IT TO CELSIUS AND RETURNS THE RESULT.THE CODE SETS AN INPUT FAHRENHEIT TEMPERATURE (100) CALLS THE FUNCTION TO GET THE CELSIUS EQUIVALENT AND THEN PRINTS BOTH THE INPUT AND THE FORMATTED OUTPUT.

Task 3: FEW-SHOT: GIVE 2–3 EXAMPLES TO CREATE A FUNCTION THAT EXTRACTS THE DOMAIN NAME FROM AN EMAIL ADDRESS

PROMPT : write a python code to create a function that extracts the domain name from an email address for example manoj@gmail.com the domain name is gmail.com,2403a52078@sru.edu.in the domain name is sru.edu.in.write a clean code.it should take input from the user and do not show examples in the output.

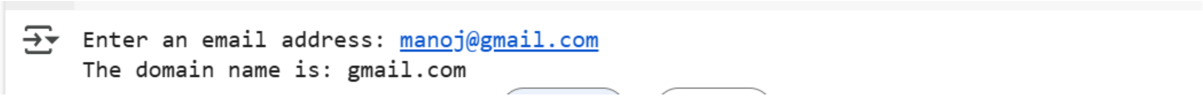
CODE GENERATED:



```python
def extract_domain(email):
  """Extracts the domain name from an email address."""
  try:
    return email.split('@')[1]
  except IndexError:
    return "Invalid email format"

# Get input from the user
email_address = input("Enter an email address: ")
domain_name = extract_domain(email_address)
print(f"The domain name is: {domain_name}")
```

OUTPUT:



```
Enter an email address: manoj@gmail.com
The domain name is: gmail.com
```

EXPLANATION: This code defines a function extract_domain that splits an email address string at the '@' symbol and returns the part after it as the domain name. It also includes error handling for invalid email

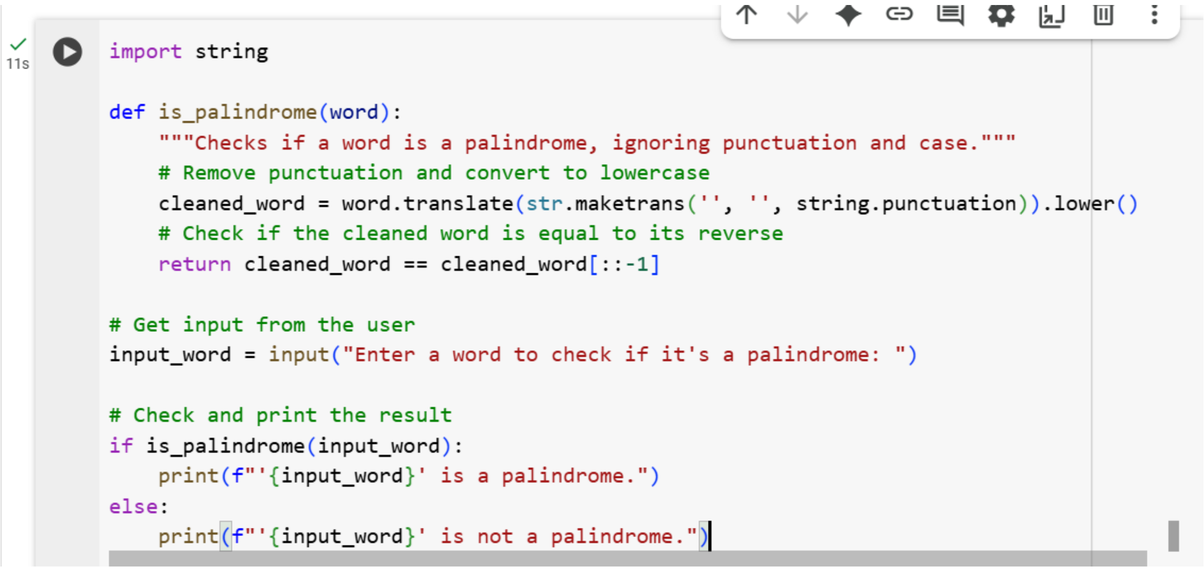
formats. The code then prompts the user for an email address, calls the function, and prints the extracted domain name.

Task 4: COMPARE ZERO-SHOT VS FEW-SHOT PROMPTING FOR GENERATING A FUNCTION THAT CHECKS WHETHER A WORD IS A PALINDROME, IGNORING PUNCTUATION AND CASE.

ZERO-SHOT:

PROMPT : Generate a python code that checks whether the given word is a palindrome or not.it should take the input from user, ignoring punctuation and case.

CODE GENERATED:



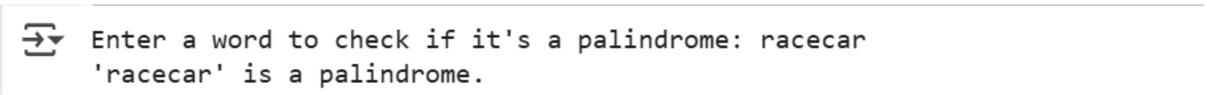
```python
import string

def is_palindrome(word):
    """Checks if a word is a palindrome, ignoring punctuation and case."""
    # Remove punctuation and convert to lowercase
    cleaned_word = word.translate(str.maketrans('', '', string.punctuation)).lower()
    # Check if the cleaned word is equal to its reverse
    return cleaned_word == cleaned_word[::-1]

# Get input from the user
input_word = input("Enter a word to check if it's a palindrome: ")

# Check and print the result
if is_palindrome(input_word):
    print(f"'{input_word}' is a palindrome.")
else:
    print(f"'{input_word}' is not a palindrome.")
```

OUTPUT:



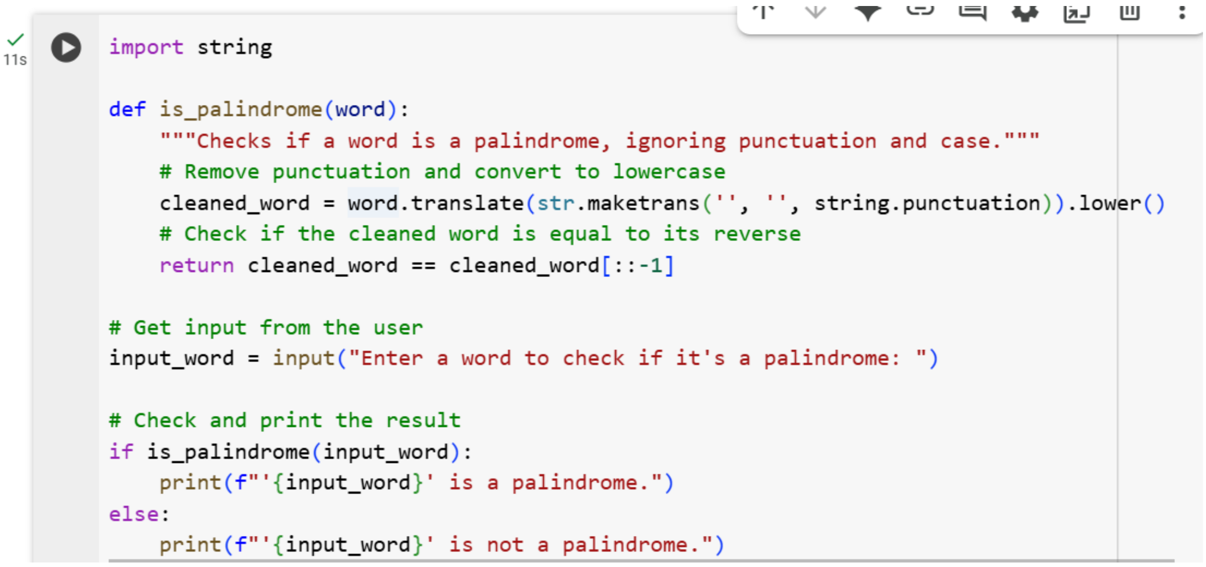
```
Enter a word to check if it's a palindrome: racecar
'racecar' is a palindrome.
```

EXPLANATION: This code defines a function is_palindrome that checks if a word is a palindrome by removing punctuation. converting it to lowercase and comparing it to its reversed version. It then takes a word as input from the user and prints whether it's a palindrome or not.

FEW-SHOT:

PROMPT : Generate a python code that checks whether the given word is a palindrome or not.it should take the input from user, ignoring punctuation and case. for example racecar is a palindrome and manideep is not a palindrome.

CODE GENERATED:



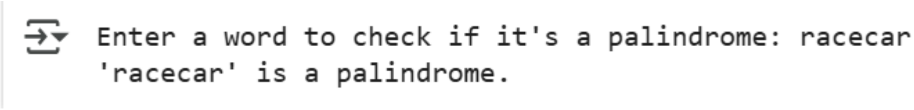
```python
import string

def is_palindrome(word):
    """Checks if a word is a palindrome, ignoring punctuation and case."""
    # Remove punctuation and convert to lowercase
    cleaned_word = word.translate(str.maketrans('', '', string.punctuation)).lower()
    # Check if the cleaned word is equal to its reverse
    return cleaned_word == cleaned_word[::-1]

# Get input from the user
input_word = input("Enter a word to check if it's a palindrome: ")

# Check and print the result
if is_palindrome(input_word):
    print(f"'{input_word}' is a palindrome.")
else:
    print(f"'{input_word}' is not a palindrome.")
```

OUTPUT:



```
Enter a word to check if it's a palindrome: racecar
'racecar' is a palindrome.
```

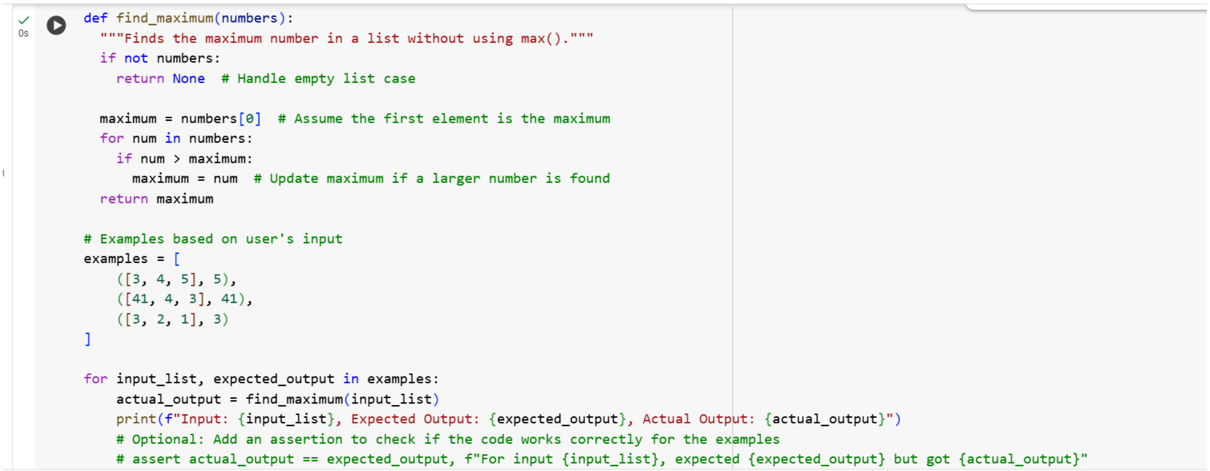
Observation: I have observed that both codes are same but I like first

Task 5: USE FEW-SHOT PROMPTING WITH 3 SAMPLE INPUTS TO GENERATE A FUNCTION THAT DETERMINES THE MAXIMUM OF THREE NUMBERS WITHOUT USING THE BUILT-IN MAX() FUNCTION.

PROMPT :

write a python code to find the maximum number of given numbers without using the built-in max() function,for example 3,4,5 the largest number is 5.41,4,3 the largest number is 41 and 3,2,1 the largest number is 3 by taking the given examples provide a perfect code by taking the examples given by me.

CODE GENERATED:



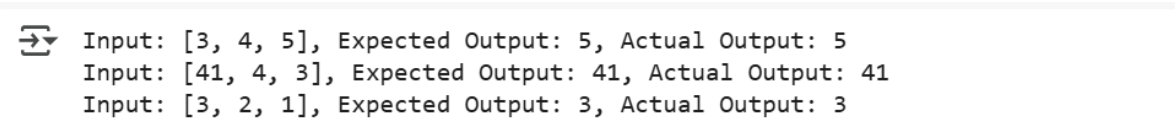
```python
def find_maximum(numbers):
  """Finds the maximum number in a list without using max()."""
  if not numbers:
    return None  # Handle empty list case

  maximum = numbers[0]  # Assume the first element is the maximum
  for num in numbers:
    if num > maximum:
      maximum = num  # Update maximum if a larger number is found
  return maximum

# Examples based on user's input
examples = [
    ([3, 4, 5], 5),
    ([41, 4, 3], 41),
    ([3, 2, 1], 3)
]

for input_list, expected_output in examples:
    actual_output = find_maximum(input_list)
    print(f"Input: {input_list}, Expected Output: {expected_output}, Actual Output: {actual_output}")
    # Optional: Add an assertion to check if the code works correctly for the examples
    # assert actual_output == expected_output, f"For input {input_list}, expected {expected_output} but got {actual_output}"
```

OUTPUT:



```
Input: [3, 4, 5], Expected Output: 5, Actual Output: 5
Input: [41, 4, 3], Expected Output: 41, Actual Output: 41
Input: [3, 2, 1], Expected Output: 3, Actual Output: 3
```

EXPLANATION:

THIS CODE DEFINES A FUNCTION FIND_MAXIMUM THAT ITERATES THROUGH A LIST OF NUMBERS TO FIND THE LARGEST ONE WITHOUT USING THE BUILT-IN MAX() FUNCTION. IT HANDLES EMPTY LISTS AND COMPARES EACH NUMBER TO A RUNNING MAXIMUM. THE CODE THEN DEMONSTRATES THE FUNCTION WITH THE EXAMPLES YOU PROVIDED, PRINTING THE INPUT, EXPECTED OUTPUT, AND ACTUAL OUTPUT FOR EACH.

-- THANK YOU--