

Python Assignment

Questions

1. File Analysis: Count Words, Lines, and Characters

Objective:

Write a Python script that reads a text file and computes:

- A. The total number of lines.
- B. The total number of words.
- C. The total number of characters.

Requirements:

- A. Accept the file path as input.
- B. Handle cases where the file does not exist gracefully.

2. Generate Fibonacci Sequence up to N and Find Primes

Objective:

Write a Python function that:

- A. Generates the Fibonacci sequence up to a given number N.
- B. Filters and prints only the prime numbers from the sequence.

3. Find the Second Largest Number in a List

Objective:

Write a Python function that takes a list of integers as input and returns the second largest number in the list.

Requirements:

- A. Handle cases where the list has duplicates.
- B. If the list has fewer than two unique numbers, return None.

4. Check If a String is a Valid Anagram

Objective: Write a Python function that checks whether two strings are anagrams of each other.

Requirements:

- A. Ignore case sensitivity.
- B. Consider only alphanumeric characters (ignore spaces and punctuation)

Answers

1.

```
def compute_file(file_path):
    try:
        with open(file_path, 'r') as file:
            lines = file.readlines()
            total_lines = len(lines)
            total_words = sum(len(line.split()) for line in lines)
            total_chars = sum(len(line) for line in lines)

            print(f"Lines--> {total_lines}")
            print(f"Words--> {total_words}")
            print(f"Chars--> {total_chars}")
    except FileNotFoundError:
        print("file doesn't exist")

file_path = input("input file path -->")
compute_file(file_path)
```

2.

```
def is_prime(num):
    if num < 2:
        return False
    for i in range(2, int(num**0.5) + 1):
        if num % i == 0:
            return False
    return True

def fibo_and_primes(n):
    fibo = []
    a, b = 0, 1
    while a <= n:
```

```

        fibo.append(a)
        a, b = b, a + b

    primes = [num for num in fibo if is_prime(num)]
    print(f"Fibonacci upto {n}: {fibo}")
    print(f"Prime Numbers in the Sequence: {primes}")

n = int(input("Enter a number N: "))
fib_and_primes(n)

```

3.

```

def second_largest(nums):
    unique_nums = list(set(nums)) # Remove duplicates
    if len(unique_nums) < 2:
        return None
    unique_nums.sort(reverse=True)
    return unique_nums[1]

nums = list(map(int, input("Enter Numbers--> ").split()))
result = second_largest(nums)
if result is None:
    print("The list has less than 2 unique numbers")
else:
    print(f"The second largest number is: {result}")

```

4.

```

import re

def valid_anagram(str1, str2):
    # Remove irrelevant characters and convert to lowercase
    clean_str1 = ''.join(re.findall(r'\w', str1)).lower()
    clean_str2 = ''.join(re.findall(r'\w', str2)).lower()

    # Check if sorted characters are equal
    return sorted(clean_str1) == sorted(clean_str2)

str1 = input("Enter string1--> ")
str2 = input("Enter string2--> ")
if valid_anagram(str1, str2):
    print("anagrams")
else:
    print("not anagrams")

```