Rajalakshmi Engineering College

Name: ANIRUDH R
Email: 240701039@rajalakshmi.edu.in
Roll no: 240701039
Phone: 9363540767
Branch: REC
Department: I CSE FA
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23221_Python Programming

REC_Python_Week 2_CY

Attempt : 2
Total Mark : 40
Marks Obtained : 37.5

Section 1 : Coding

1. Problem Statement

Max is fascinated by prime numbers and the Fibonacci sequence. He wants to combine these two interests by creating a program that outputs the first n prime numbers within the Fibonacci sequence.

Your task is to help Max by writing a program that prints the first n prime numbers in the Fibonacci sequence using a while loop along with the break statement to achieve the desired functionality.

Input Format

The input consists of an integer n, representing the number of prime Fibonacci numbers to generate.

Output Format

The output displays space-separated first n prime numbers found in the Fibonacci sequence.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 5

Output: 2 3 5 13 89

Answer

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

def fibonacci_primes(n):
    primes = []
    a,b=0,1
    while len(primes)<n:
        fib_num = a+b
        if is_prime(fib_num):
            primes.append(fib_num)
        a,b=b,fib_num
    return primes
n = int(input())
result = fibonacci_primes(n)
print(*result)
```

Status : Correct ***Marks : 10/10***

2. Problem Statement

Nisha is a mathematics enthusiast, eager to explore the realm of twin prime numbers. The objective is to develop a program that enables the

discovery and presentation of twin prime pairs.

The program should take an integer 'n' as input and generate 'n' pairs of twin primes, displaying the pairs with a difference of 2 between them.

Input Format

The input consists of a single integer, n.

Output Format

The output displays the 'n' pairs of twin primes, the pairs with a difference of 2 between them.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 5

Output: 3 5
5 7
11 13
17 19
29 31

Answer

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

def find_twin_primes(n):
    twin_primes = []
    num = 2
    while len(twin_primes)<n:
        if is_prime(num) and is_prime(num + 2):
            twin_primes.append((num,num + 2))
        num+=1
```

```
    return twin_primes

n = int(input())
result = find_twin_primes(n)
for pair in result:
    print(pair[0],pair[1])
```

Status : Correct **Marks : 10/10**

3. Problem Statement

Rohith is a data analyst who needs to categorize countries based on their population growth rates. Each country is assigned a unique code. Rohith will receive a code and corresponding data based on the code. If the data falls within specific thresholds, he needs to classify the country's priority level.

Your task is to write a program that reads a country code and its associated data, and then determines if the priority is "High" or "Low."

Thresholds:France: Priority is "High" if the percentage < 50, else "Low".Japan: Priority is "High" if life expectancy > 80, else "Low".Brazil: Priority is "High" if the urban population > 80, else "Low".

Input Format

The first line of input consists of an integer, representing the country code (1 for France, 2 for Japan, 3 for Brazil).

If the country code is 1,

- The second line consists of a floating-point value N, representing the percentage of the English-speaking population.

If the country code is 2,

- The second line consists of a floating-point value A, representing the average life expectancy in years.

If the country code is 3,

- The second line consists of a floating-point value P, representing the percentage of the urban population.

Output Format

The first line of output displays "Priority: High" or "Priority: Low" based on the input data.

If the country code is invalid, print "Invalid".

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 1
30.0

Output: Priority: High

Answer

```
def classify_priority(country_code,data):
    if country_code == 1:
        percentage = data
        if percentage<50:
            return "High"
        else:
            return "Low"
    elif country_code == 2:
        life_expectancy = data
        if life_expectancy > 80:
            return "High"
        else:
            return "Low"
    elif country_code == 3:
        urban_population = data
        if urban_population > 80:
            return "High"
        else:
            return "Low"

    else:
        return"Invalid "
```

```
country_code = int(input())
data = float(input())
priority = classify_priority(country_code,data)
if priority == "Invalid":
   print(priority)
else:
      print("priority:",priority)
```

Status : Partially correct ***Marks : 7.5/10***

4. Problem Statement

Alex is practicing programming and is curious about prime and non-prime digits. He wants to write a program that calculates the sum of the non-prime digits in a given integer using loops.

Help Alex to complete his task.

Example:

Input:

845

output:

12

Explanation:

Digits: 8 (non-prime), 4 (non-prime), 5 (prime)

The sum of Non-Prime Digits: 8 + 4 = 12

Output: 12

Input Format

The input consists of a single integer X.

Output Format

The output prints an integer representing the sum of non-prime digits in X.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 845

Output: 12

Answer

```
def is_prime_digit(digit):
    if digit<=1:
        return False
    if digit<=3:
        return True
    if digit== 5 or digit == 7:
        return True
    return False

def sum_non_prime_digits(number):
    number_str = str(number)
    non_prime_sum = 0
    for digit_char in number_str:
        digit = int(digit_char)
        if not is_prime_digit(digit):
            non_prime_sum += digit
    return non_prime_sum

number = int(input())
result = sum_non_prime_digits(number)
print(result)
```

Status : Correct ***Marks : 10/10***