Rajalakshmi Engineering College

Name: Tanisha Sudhagar
Email: 240701553@rajalakshmi.edu.in
Roll no: 2116240701553
Phone: 6374776137
Branch: REC
Department: I CSE FF
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23221_Python Programming

REC_Python_Week 2_CY

Attempt : 1
Total Mark : 40
Marks Obtained : 40

Section 1 : Coding

1. Problem Statement

Students are allowed to work on our computer center machines only after entering the correct secret code. If the code is correct, the message "Logged In" is displayed. They are not allowed to log in to the machine until they enter the correct secret code.

Write a program to allow the student to work only if he/she enters the correct secret code.

Note: Here, secret code means the last three digits should be divisible by the first digit of the number.

Input Format

The input consists of an integer n, which represents the secret code.

Output Format

The output displays either "Logged In" or "Incorrect code" based on the given condition.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 2345

Output: Incorrect code

Answer

```
# You are using Python
n=int(input())
first_digit=int(str(n)[0])
last_three=n%1000
if last_three%first_digit==0:
    print("Logged In")
else:
    print("Incorrect Code")
```

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

```
# You are using Python
def is_prime(p):
    if p<=1:
        return False
    if p==2:
        return True
    if p%2==0:
        return False
    for i in range(3,int(p**0.5)+1,2):
        if p%i==0:
            return False
    return True
n= int(input())
count=0
current=3
while count<n:
    if is_prime(current) and is_prime(current+2):
        print(f"{current} {current+2}")
        count+=1
    current+=2
```

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

3. 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

```
# You are using Python
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 fibonacci_prime(n):
    fib1,fib2=0,1
    count=0

    while count<n:
```

```
        if is_prime(fib1):
            print(fib1,end=" ")
            count+=1
        fib1,fib2=fib2,fib1+fib2

n=int(input())
fibonacci_prime(n)
```

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

4. Problem Statement

Gabriel is working on a wildlife research project where he needs to compute various metrics for different animals based on their characteristics. Each animal type requires a different calculation: a deer's distance traveled, a bear's weight based on footprint size, or a bird's altitude based on its flying pattern.

Conditions:

For Deer (Mode 'D' or 'd'): Distance = speed of sound * time taken, where the speed of sound in air is 343 meters per second.For Bear (Mode 'B' or 'b'): Weight = footprint size * average weight, where the average weight per square inch for a bear is 5.0 pounds.For Bird (Mode 'F' or 'f'): Altitude = flying pattern * distance covered (in meters).

Write a program to help Gabriel analyze the characteristics of animals based on the given inputs.

Input Format

The first line of input consists of a character, representing the type of animal 'D/d' for deer, 'B/b' for bear, and 'F/f' for bird.

If the choice is 'D' or 'd':

The second line of input consists of a floating-point value T, representing the

time taken from the deer's location to the observer.

If the choice is 'B' or 'b':

The second line of input consists of a floating-point value S, representing the size of the bear's footprint in square inches.

If the choice is 'F' or 'f':

1. The second line of input consists of a floating-point value P, representing the bird's flying pattern.
2. The third line consists of a floating-point value D, representing the distance covered by the bird in meters.

Output Format

The output prints one of the following:

If the choice is 'D' or 'd':

The output prints "Distance: X m" where X is a floating point value rounded off to two decimal places, representing the calculated distance traveled by the sound wave in meters.

If the choice is 'B' or 'b':

The output prints "Weight: Y lb" where Y is a floating point value rounded off to two decimal places, representing the estimated weight of the bear in pounds.

If the choice is 'F' or 'f':

The output prints "Altitude: Z m" where Z is a floating point value rounded off to two decimal places, representing the calculated altitude of the bird's flight in meters.

If the given choice is invalid, print "Invalid".

Refer to the sample output for formatting specifications.

Sample Test Case

Input: d
2.5
Output: Distance: 857.50 m

Answer

```
def deer_distance(time):
    return 343*time
def bear_weight(footprint):
    return 5.0*footprint
def bird_altitude(fly_pattern,distance):
    return fly_pattern*distance

mode=input().strip().lower()
if mode=='d':
    time=float(input())
    print("Distance:{:.2f} m".format(deer_distance(time)))
elif mode=='b':
    footprint=float(input())
    print("Weight:{:.2f} lb".format(bear_weight(footprint)))
elif mode=='f':
    fly_pattern=float(input())
    distance=float(input())
    print("Altitude:{:.2f} m".format(bird_altitude(fly_pattern,distance)))
else:
    print("Invalid")
```

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