Rajalakshmi Engineering College

Name: Devadharshini K
Email: 240701626@rajalakshmi.edu.in
Roll no: 2116240701626
Phone: 8248408481
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 : 2
Total Mark : 40
Marks Obtained : 35

Section 1 : Coding

1. 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

```
# Read the input integer
X = int(input())

# Initialize sum of non-prime digits
sum_non_prime = 0

# Convert the number to string to iterate through each digit
for digit_char in str(X):
    digit = int(digit_char)
    # Check if the digit is non-prime
    if digit in [0, 1, 4, 6, 8, 9]:
        sum_non_prime += digit

# Print the result
print(sum_non_prime)
```

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

2. 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

```
# Read the animal type character
animal_type = input().strip()

# Check if the input is valid
if animal_type in ['D', 'd']:
    # Read the time taken for deer
    T = float(input())
    # Calculate distance traveled by sound
    distance = 343 * T
    print(f"Distance: {distance:.2f} m")
```

```
elif animal_type in ['B', 'b']:
    # Read the footprint size for bear
    S = float(input())
    # Calculate weight
    weight = S * 5.0
    print(f"Weight: {weight:.2f} lb")
elif animal_type in ['F', 'f']:
    # Read the flying pattern and distance for bird
    P = float(input())
    D = float(input())
    # Calculate altitude
    altitude = P * D
    print(f"Altitude: {altitude:.2f} m")
else:
    # Invalid input
    print("Invalid")
```

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

3. Problem Statement

Taylor is tasked with a mathematical challenge that requires finding the smallest positive number divisible by all integers from 1 to n.

Help Taylor to determine the smallest positive number that is divisible by all integers from 1 to n. Make sure to employ the break statement to ensure efficiency in the program.

Input Format

The input consists of a single integer, n.

Output Format

The output displays the smallest positive number that is divisible by all integers from 1 to n.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 10

Output: 2520

Answer

```
n = int(input())

# Initialize candidate number to n (since it must be at least n)
candidate = n

while True:
    for i in range(1, n + 1):
        if candidate % i != 0:
            candidate += 1
            break  # Exit the loop early if not divisible
    else:
        # If the loop completes without break, candidate is divisible by all
        print(candidate)
        break
```

Status : Partially correct ***Marks : 5/10***

4. 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 < 2:
        return False
    for i in range(2, int(num**0.5) + 1):
        if num % i == 0:
            return False
    return True

n = int(input())

count = 0
a, b = 0, 1  # Starting Fibonacci numbers
result = []

while True:
    # Generate next Fibonacci number
    a, b = b, a + b

    # Check if Fibonacci number is prime
    if is_prime(a):
        result.append(a)
        count += 1
        if count == n:
            break  # Found all required prime Fibonacci numbers

# Print space-separated prime Fibonacci numbers
print(' '.join(map(str, result)))
```

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