

# Rajalakshmi Engineering College

Name: Rupakanthan A  
Email: 240701444@rajalakshmi.edu.in  
Roll no: 240701444  
Phone: 9342869897  
Branch: REC  
Department: I CSE FE  
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

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**

```
import math
def smallest_multiple(n):
    result = 1
    for i in range(2, n + 1):
        result = (result * i) // math.gcd(result, i)
    return result
n = int(input())
print("\n",smallest_multiple(n))
```

**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 < 2:  
        return False  
    for j in range(2, int(num**0.5) + 1):  
        if num % j == 0:  
            return False  
    return True  
n = int(input())  
count = 0  
i = 3  
while count < n:  
    if is_prime(i) and is_prime(i + 2):  
        print(i, i + 2)  
        count += 1  
    i += 1
```

**Status :** Correct

**Marks :** 10/10

### 3. 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***

```
n = int(input())
code_str = str(n)
```

```
first_digit = int(code_str[0])
last_three_digits = int(code_str[-3:])
```

```
if first_digit != 0 and last_three_digits % first_digit == 0:
    print("Logged In")
else:
    print("Incorrect code")
```

**Status :** Correct

**Marks :** 10/10

## **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***

```
x = int(input())
sum_non_prime = 0
prime_digits = {2,3,5,7}
```

```
for digit in str(x):
    d = int(digit)
```

```
if d not in prime_digits:  
    sum_non_prime += d  
print(sum_non_prime)
```

**Status :** Correct

**Marks :** 10/10