LEVEL WISE QUESTIONS(Sample2)

1)Write a program to print all the Non-Prime numbers between A and B?

PROGRAM:-

```
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 non_prime_numbers(A, B):
    non_primes = []
    for num in range(A, B+1):
        if not is_prime(num):
            non_primes.append(num)
    return non_primes


A = 12
B = 19
result = non_prime_numbers(A, B)
print("Non-prime numbers between", A, "and", B, "are:", ", ".join(map(str, result)))
```

Output

```
Non-prime numbers between 12 and 19 are: 12, 14, 15, 16, 18
```

2)Find the year of the given Anniversary is leap year or not. If leap year then print the next Anniversary, if not leap year then print the previous Anniversary.

from datetime import datetime

PROGRAM:-

```
def is_leap_year(year):
    # Check if the year is a leap year
    if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):
        return True
    else:
        return False


def anniversary_date(date_str):
    # Convert the date string to a datetime object
    date_obj = datetime.strptime(date_str, "%d/%m/%Y")
    anniversary_year = date_obj.year


    if is_leap_year(anniversary_year):
        # If leap year, calculate next anniversary
        next_anniversary = datetime(anniversary_year + 4, date_obj.month, date_obj.day)
        print(f"Given Anniversary Year: Leap Year. Anniversary Date:
{next_anniversary.strftime('%d/%m/%Y')}")
    else:
        # If not leap year, calculate previous anniversary
        previous_anniversary = datetime(anniversary_year - 1, date_obj.month, date_obj.day)
        print(f"Given Anniversary Year: Non Leap Year. Anniversary Date:
{previous_anniversary.strftime('%d/%m/%Y')}")


# Input
date_input = input("Enter Date (DD/MM/YYYY): ")
anniversary_date(date_input)
```

```
Output
Enter Date (DD/MM/YYYY): 04/11/1947
Given Anniversary Year: Non Leap Year. Anniversary Date: 04/11/1946
```

3)Write a program to print the given number is Perfect number or not?

PROGRAM:-

```
def is_perfect_number(num):
    # Calculate the sum of divisors of the number
    sum_of_divisors = 0
    for i in range(1, num):
        if num % i == 0:
            sum_of_divisors += i

    # Check if the sum of divisors equals the number
    if sum_of_divisors == num:
        return True
    else:
        return False

# Input
num = int(input("Given Number: "))

# Check if it's a perfect number
if is_perfect_number(num):
    print("Its a Perfect Number")
else:
    print("Its not a Perfect Number")
```

Output

```
Given Number: 6
Its a Perfect Number
```

4)Write a program to generate Pythagorean Triplets for the given limit.

PROGRAM:-

```
def generate_pythagorean_triplets(limit):
    # Generate Pythagorean triplets (a, b, c) where a^2 + b^2 = c^2
    triplets = []
    for a in range(1, limit):
        for b in range(a, limit):  # Start b from a to avoid duplicate triplets like (3, 4, 5) and (4, 3, 5)
            c = (a**2 + b**2) ** 0.5  # Calculate c using the Pythagorean theorem
            if c.is_integer():  # Check if c is an integer
                triplets.append((a, b, int(c)))

    return triplets

# Input: upper limit
limit = int(input("Enter upper limit: "))

# Generate and print Pythagorean triplets
triplets = generate_pythagorean_triplets(limit)
for triplet in triplets:
    print(*triplet)
```

Output

```
Enter upper limit: 10
3 4 5
6 8 10
```

5)Write a program to find the sum of digits of N digit number (sum should be single digit)

PROGRAM:-

```
def sum_of_digits(num):
    # Sum the digits of the number
    while num >= 10:  # Repeat until the sum is a single digit
        num = sum(int(digit) for digit in str(num))  # Sum the digits
    return num

# Input: N value and the N-digit number
n = int(input("Enter N value: "))
number = int(input(f"Enter {n} digit number: "))

# Calculate the sum of digits until it's a single digit
result = sum_of_digits(number)

# Output the result
print(f"Sum of {n} digit number: {result}")
```

Output

```
Enter N value: 3
Enter 3 digit number: 143
Sum of 3 digit number: 8
```

6)Program to find whether the given number is Armstrong number or not

PROGRAM:-

```
def is_armstrong(number):
    # Convert the number to string to calculate the number of digits
    num_str = str(number)
    num_digits = len(num_str)
```

```
    # Calculate the sum of digits raised to the power of the number of digits
    sum_of_powers = sum(int(digit) ** num_digits for digit in num_str)

    # Check if the sum is equal to the original number
    return sum_of_powers == number

# Input
number = int(input("Enter number: "))

# Check if the number is Armstrong
if is_armstrong(number):
    print("Given number is Armstrong number")
else:
    print("Given number is not Armstrong number")
```

```
Output

Enter number: 153
Given number is Armstrong number
```

7)Program to find whether the given number is Harshad number or not

PROGRAM:-

```
def is_harshad(number):
    # Calculate the sum of the digits of the number
    digit_sum = sum(int(digit) for digit in str(number))

    # Check if the number is divisible by the sum of its digits
    return number % digit_sum == 0

# Input
```

```
number = int(input("Enter number: "))

# Check if the number is Harshad
if is_harshad(number):
    print("Given number is Harshad number")
else:
    print("Given number is not Harshad number")
```

```
Output

Enter number: 21
Given number is Harshad number
```

8)Program to find whether the given number is Happy number or not

PROGRAM:-

```
def sum_of_squares_of_digits(number):
    return sum(int(digit) ** 2 for digit in str(number))

def is_happy_number(number):
    seen = set()  # To track numbers we've already seen to avoid infinite loops
    while number != 1 and number not in seen:
        seen.add(number)
        number = sum_of_squares_of_digits(number)
    return number == 1

# Input
number = int(input("Enter number: "))

# Check if the number is Happy
if is_happy_number(number):
    print("Given number is happy number")
```

```
else:
    print("Given number is not happy number")
```

```
Output

Enter number: 19
Given number is happy number
```

9)Program to find whether the given number is Tech number or not

PROGRAM:-

```
def is_tech_number(number):
    num_str = str(number)
    num_len = len(num_str)

    # Check if the number has an even number of digits
    if num_len % 2 != 0:
        return False

    # Split the number into two halves
    mid = num_len // 2
    first_half = int(num_str[:mid])
    second_half = int(num_str[mid:])

    # Check if the sum of squares of both halves equals the original number
    if (first_half**2 + second_half**2) == number:
        return True
    else:
        return False

# Input
number = int(input("Enter number: "))
```

```
# Check if the number is a Tech number
if is_tech_number(number):
    print("Given number is Tech number")
else:
    print("Given number is not Tech number")
```

```
Output

Enter number: 3025
Given number is not Tech number
```

10)Write a program using function to calculate the simple interest. Suppose the customer is a senior citizen. She is being offered 15 percent rate of interest; he is being offered 12 percent rate of interest for all other customers, the ROI is 10 percent.

PROGRAM:-

```
def calculate_simple_interest(principal, years, rate_of_interest):
    # Simple Interest formula: SI = (P * R * T) / 100
    return (principal * rate_of_interest * years) / 100

def get_rate_of_interest(gender, is_senior_citizen):
    # Determine the rate of interest based on the customer type
    if is_senior_citizen.lower() == 'y':
        return 15  # 15% for senior citizens
    elif gender.lower() == 'm':
        return 12  # 12% for male customers
    else:
        return 10  # 10% for all other customers

# Input from the user
```

```
principal = float(input("Enter the principal amount: "))
years = int(input("Enter the no of years: "))
gender = input("Gender (m/f): ")
is_senior_citizen = input("Is customer senior citizen (y/n): ")

# Get the appropriate rate of interest
rate_of_interest = get_rate_of_interest(gender, is_senior_citizen)

# Calculate the simple interest
interest = calculate_simple_interest(principal, years, rate_of_interest)

# Output the interest
print(f"Interest: {interest}")
```

Output

```
Enter the principal amount: 20000
Enter the no of years: 3
Gender (m/f): M
Is customer senior citizen (y/n): N
Interest: 7200.0
```