**1. Write a Python program to check if the given number is a Disarium Number?**

**ans:** def task1(num):

    try:

        if type(num) != int:

            raise Exception("Not an integer")

        num = str(num)

        total = 0

        for i in range(0, len(num)):

            n = int(num[i])

            n = n\*\*(i+1)

            total = total + n

        if total == int(num):

            return True

        else:

            return False

    except Exception as e:

        print(e)

task1(147)

False

task1(135)

True

**2. Write a Python program to print all disarium numbers between 1 to 100?**

**ans:** for i in range(1, 101):

    if task1(i) == True:

        print(i, end = ' ')

1 2 3 4 5 6 7 8 9 89

**3. Write a Python program to check if the given number is Happy Number?**

**ans:** def task3(num):

    res = num;

    def isHappy(num):

        r = 0;

        s = 0

        while(num > 0):

            r = num%10

            s += r\*\*2

            num //= 10

        return s

    while(res != 1 and res != 4):

        res = isHappy(res)

    if(res == 1):

        return True

    elif(res == 4):

        return False

task3(100)

True

task3(4)

False

**4. Write a Python program to print all happy numbers between 1 and 100?**

**ans:** for i in range(1, 101):

    if task3(i) == True:

        print(i, end = ' ')

1 7 10 13 19 23 28 31 32 44 49 68 70 79 82 86 91 94 97 100

**5. Write a Python program to determine whether the given number is a Harshad Number?**

**ans:** def task5(num):

    temp = num

    total = 0

    while (temp > 0):

        total = total + (temp%10)

        temp = temp//10

    if num%total==0:

        print("Harshad Number")

    else:

        print("Not Harshad Number")

task5(156)

Harshad Number

task5(113)

Not Harshad Number

**6. Write a Python program to print all pronic numbers between 1 and 100?**

**ans:** i = 0

while True:

    pronicNum = i\*(i+1)

    i = i + 1

    if pronicNum > 100:

        break

    print(pronicNum, end = ' ')

0 2 6 12 20 30 42 56 72 90