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

# define the number

num = 175

# convert the number to a string to access individual digits

num\_str = str(num)

# initialize a variable to store the sum of powers of digits

sum\_powers = 0

# iterate over the digits of the number and calculate the sum of powers

for i, digit in enumerate(num\_str):

sum\_powers += int(digit)\*\*(i+1)

# check if the sum of powers is equal to the original number

if sum\_powers == num:

print(num, "is a Disarium number")

else:

print(num, "is not a Disarium number")

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

def is\_disarium(num):

# convert the number to a string to access individual digits

num\_str = str(num)

# initialize a variable to store the sum of powers of digits

sum\_powers = 0

# iterate over the digits of the number and calculate the sum of powers

for i, digit in enumerate(num\_str):

sum\_powers += int(digit)\*\*(i+1)

# check if the sum of powers is equal to the original number

if sum\_powers == num:

return True

else:

return False

# iterate over numbers from 1 to 100 and print the Disarium numbers

for num in range(1, 101):

if is\_disarium(num):

print(num)

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

def is\_happy(num):

# initialize a set to keep track of visited numbers

visited = set()

while True:

# convert the number to a string to access individual digits

num\_str = str(num)

# initialize a variable to store the sum of squares of digits

sum\_squares = 0

# iterate over the digits of the number and calculate the sum of squares

for digit in num\_str:

sum\_squares += int(digit)\*\*2

# check if the sum of squares is 1

if sum\_squares == 1:

return True

# check if the sum of squares has been visited before

if sum\_squares in visited:

return False

# add the sum of squares to the visited set and set it as the new number

visited.add(sum\_squares)

num = sum\_squares

# test the function with some examples

print(is\_happy(19)) # should print True

print(is\_happy(4)) # should print False

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

def is\_happy(num):

# initialize a set to keep track of visited numbers

visited = set()

while True:

# convert the number to a string to access individual digits

num\_str = str(num)

# initialize a variable to store the sum of squares of digits

sum\_squares = 0

# iterate over the digits of the number and calculate the sum of squares

for digit in num\_str:

sum\_squares += int(digit)\*\*2

# check if the sum of squares is 1

if sum\_squares == 1:

return True

# check if the sum of squares has been visited before

if sum\_squares in visited:

return False

# add the sum of squares to the visited set and set it as the new number

visited.add(sum\_squares)

num = sum\_squares

# loop through the numbers from 1 to 100 and check if they are Happy numbers

for num in range(1, 101):

if is\_happy(num):

print(num)

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

def is\_harshad(num):

# calculate the sum of the digits

digit\_sum = sum(int(digit) for digit in str(num))

# check if the number is divisible by the sum of its digits

if num % digit\_sum == 0:

return True

else:

return False

# get user input for the number to check

num = int(input("Enter a number: "))

# check if the number is a Harshad Number and print the result

if is\_harshad(num):

print(num, "is a Harshad Number.")

else:

print(num, "is not a Harshad Number.")

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

def is\_pronic(num):

# iterate through the range from 1 to the square root of the number

for i in range(1, int(num\*\*0.5) + 1):

# check if the product of the two consecutive integers equals the number

if i \* (i + 1) == num:

return True

return False

# iterate through the range from 1 to 100

for i in range(1, 101):

# check if the number is a pronic number and print it if it is

if is\_pronic(i):

print(i)