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

def is\_disarium\_number(number):

# Convert the number to a string to access individual digits

num\_str = str(number)

length = len(num\_str)

sum\_of\_digits = 0

# Calculate the sum of digits raised to their respective positions

for i in range(length):

digit = int(num\_str[i])

sum\_of\_digits += digit \*\* (i + 1)

# Check if the sum of digits is equal to the original number

if sum\_of\_digits == number:

return True

else:

return False

# Example usage

number = 135

if is\_disarium\_number(number):

print(number, "is a Disarium number")

else:

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

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

def is\_disarium\_number(number):

# Convert the number to a string to access individual digits

num\_str = str(number)

length = len(num\_str)

sum\_of\_digits = 0

# Calculate the sum of digits raised to their respective positions

for i in range(length):

digit = int(num\_str[i])

sum\_of\_digits += digit \*\* (i + 1)

# Check if the sum of digits is equal to the original number

if sum\_of\_digits == number:

return True

else:

return False

# Print Disarium numbers between 1 to 100

print("Disarium numbers between 1 to 100:")

for number in range(1, 101):

if is\_disarium\_number(number):

print(number)

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

def is\_happy\_number(number):

visited = set()

while True:

# Calculate the sum of squares of digits

sum\_of\_squares = sum(int(digit) \*\* 2 for digit in str(number))

# Check if the sum is 1, indicating a happy number

if sum\_of\_squares == 1:

return True

# Check if the sum has been visited before, indicating a cycle

if sum\_of\_squares in visited:

return False

# Add the sum to visited set and continue with the next iteration

visited.add(sum\_of\_squares)

number = sum\_of\_squares

# Test the function

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

if is\_happy\_number(number):

print(number, "is a Happy Number")

else:

print(number, "is not a Happy Number")

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

def is\_happy\_number(number):

visited = set()

while True:

# Calculate the sum of squares of digits

sum\_of\_squares = sum(int(digit) \*\* 2 for digit in str(number))

# Check if the sum is 1, indicating a happy number

if sum\_of\_squares == 1:

return True

# Check if the sum has been visited before, indicating a cycle

if sum\_of\_squares in visited:

return False

# Add the sum to visited set and continue with the next iteration

visited.add(sum\_of\_squares)

number = sum\_of\_squares

# Print all happy numbers between 1 and 100

print("Happy numbers between 1 and 100:")

for number in range(1, 101):

if is\_happy\_number(number):

print(number)

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

def is\_harshad\_number(number):

# Calculate the sum of digits

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

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

if number % digit\_sum == 0:

return True

else:

return False

# Take input from the user

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

# Check if the number is a Harshad Number

if is\_harshad\_number(number):

print(number, "is a Harshad Number")

else:

print(number, "is not a Harshad Number")

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

def is\_pronic\_number(number):

# Find the square root of the number

square\_root = int(number \*\* 0.5)

# Check if the product of two consecutive integers is equal to the number

if square\_root \* (square\_root + 1) == number:

return True

else:

return False

# Print all pronic numbers between 1 and 100

for num in range(1, 101):

if is\_pronic\_number(num):

print(num)