SIMATS ENGINEERING

SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES

CHENNAI-602105

CSA0839 – PYTHON PROGRAMMING FOR CYBER SECURITY

Submitted By :S. Jaswanth Venkata Rahul

# 1. Non-prime numbers between A and B

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

b = int(input("Enter B: "))

for x in range(a, b+1):

if x > 1:

for i in range(2, x):

if x % i == 0:

print(x, end=' ')

break

print()

# 2. Leap year check and Anniversary adjustment

date = input("Enter the date to be checked (DD/MM/YYYY): ")

c = date.split("/")

b = list(map(int, c))

input\_year = b[2]

if input\_year % 4 == 0:

if input\_year % 100 == 0:

if input\_year % 400 == 0:

print("%d is Leap Year" % input\_year)

else:

print("%d is not the Leap Year" % input\_year)

else:

print("%d is the Leap Year" % input\_year)

else:

print("%d is not the Leap Year" % input\_year)

x = input\_year % 4

if x != 0:

print("Previous Leap year:", input\_year - x)

else:

print("Next leap year:", input\_year + 4)

# 3. Perfect Number

Number = int(input("Enter any Number: "))

Sum = 0

for i in range(1, Number):

if Number % i == 0:

Sum = Sum + i

if Sum == Number:

print("%d is a Perfect Number" % Number)

else:

print("%d is not a Perfect Number" % Number)

# 4. Generate Pythagorean Triplets

A = input("Enter upper limit: ")

c = 0

m = 2

if A.isnumeric():

x = int(A)

while c < x:

for n in range(1, m + 1):

a = m \* m - n \* n

b = 2 \* m \* n

c = m \* m + n \* n

if c > x:

break

if a == 0 or b == 0 or c == 0:

break

print(a, b, c)

m += 1

else:

print("Invalid input")

# 5. Sum of digits

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

Sum = 0

temp = num

while temp > 0:

digit = temp % 10

Sum += digit

temp //= 10

print("Sum of Digits:", Sum)

# 6. Armstrong number

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

Sum = 0

temp = num

while temp > 0:

digit = temp % 10

Sum += digit \*\* 3

temp //= 10

if num == Sum:

print("Armstrong Number")

else:

print("Not a Armstrong Number")

# 7. Harshad Number

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

Sum = 0

temp = num

while temp > 0:

digit = temp % 10

Sum += digit

temp //= 10

if num % Sum == 0:

print("Harshad Number")

else:

print("Not a Harshad Number")

# 8. Happy Number

def happy(n):

temp = n

sum = 0

while temp > 0:

digit = temp % 10

sum += digit \*\* 2

temp //= 10

return sum

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

result = num

while result != 1 and result != 4:

result = happy(result)

if result == 1:

print("True")

else:

print("False")

# 9. Tech Number

n = 3025

m = str(n)

a = len(m) // 2

b = len(m) // 2

c = int(m[:a])

d = int(m[a:])

if (c + d) \*\* 2 == n:

print("Tech Number")

else:

print("Not a Tech Number")

# 10. Simple Interest with conditions

p = int(input("Enter the Principal amount: "))

n = int(input("Enter the number of years: "))

SC = input("Senior Citizen Yes/No: ")

G = input("Male/Female: ")

if SC == 'Y' and G == 'M':

print("SI =", (p \* n \* 12) / 100)

elif SC == 'Y' and G == 'F':

print("SI =", (p \* n \* 15) / 100)

else:

print("SI =", (p \* n \* 10) / 100)

# 11 & 12. Number of factors and nth factor

x = int(input("Enter the number: "))

y = []

for i in range(1, x + 1):

if x % i == 0:

y.append(i)

print("Number of factors:", len(y))

print("Factors are:", y)

n = int(input("Enter N value: "))

if n <= len(y):

print("Nth factor is:", y[n - 1])

else:

print("Invalid N")

# 13. Unique permutations

import itertools

n = input("Enter the number: ")

P = list(itertools.permutations(n))

for i in set(P):

print("".join(i))

# 14. Square and Cube

import math

num = float(input("Enter the number: "))

print("Square Number:", math.pow(num, 2))

print("Cube Number:", round(math.pow(num, 3), 3))

# 15. Binary to Decimal/Octal/Hex

num = input("Enter the binary number: ")

bin\_num = "01"

binary = True

for i in num:

if i not in bin\_num:

binary = False

print("Invalid input")

break

if binary:

dec\_number = int(num, 2)

oct\_number = oct(dec\_number)

hexa = hex(dec\_number)

print("Decimal Equivalent:", dec\_number)

print("Octal Equivalent:", oct\_number)

print("Hexa Equivalent:", hexa)























