**9- Functions**

**Ex. No. : 9.1 Date: 29/05/2024**

**Register No: 231401037 Name: ILANKO M**

**Coin change**

complete function to implement coin change making problem i.e. finding the minimum

number of coins of certain denominations that add up to given amount of money.

The only available coins are of values 1, 2, 3, 4

**Input Format:**

Integer input from stdin.

**Output Format:**

return the minimum number of coins required to meet the given target.

**Example Input:**

16

**Output:**

4

**Explanation:**

We need only 4 coins of value 4 each

**Example Input:**

25

**Output:**

7

**Explanation:**

We need 6 coins of 4 value, and 1 coin of 1 value

def coinChange(n):

dp = [float('inf')] \* (n + 1)

dp[0] = 0

coins = [1, 2, 3, 4]

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

for coin in coins:

if i - coin >= 0:

dp[i] = min(dp[i], dp[i - coin] + 1)

return dp[n]

**For example:**

| **Test** | **Result** |
| --- | --- |
| print(coinChange(16)) | 4 |

**Ex. No. : 9.2 Date: 29/05/2024**

**Register No:231401037 Name: ILANKO M**

**Abundant number**

An abundant number is a number for which the sum of its proper divisors is greater than

the number itself. Proper divisors of the number are those that are strictly lesser than the number.

Input Format:

Take input an integer from stdin

Output Format:

Return Yes if given number is Abundant. Otherwise, print No

Example input:

12

Output:

Yes

Explanation

The proper divisors of 12 are: 1, 2, 3, 4, 6, whose sum is 1 + 2 + 3 + 4 + 6 = 16. Since sum of

proper divisors is greater than the given number, 12 is an abundant number.

Example input:

13

Output:

No

Explanation

The proper divisors of 13 is: 1, whose sum is 1. Since sum of proper divisors is not greater

than the given number, 13 is not an abundant number.

**For example:**

| **Test** | **Result** |
| --- | --- |
| print(abundant(12)) | Yes |
| print(abundant(13)) | No |

**def abundant(num):**

**divisors\_sum = sum([i for i in range(1, num) if num % i == 0])**

**if divisors\_sum > num:**

**return "Yes"**

**else:**

**return "No"**

| **Test** | **Expected** | **Got** |  |
| --- | --- | --- | --- |
|  | print(abundant(12)) | Yes | Yes |  |
|  | print(abundant(13)) | No | No |  |

**Ex. No. : 9.3 Date: 29/05/2024**

**Register No:231401037 Name: ILANKO M**

**Product at even place divisible by sum of digits**

Write a code to check whether product of digits at even places is divisible by sum of digits

at odd place of a positive integer.

Input Format:

Take an input integer from stdin.

Output Format:

Print TRUE or FALSE.

Example Input:

1256

Output:

TRUE

Example Input:

1595

Output:

FALSE

**For example:**

| **Test** | **Result** |
| --- | --- |
| print(productDigits(1256)) | True |
| print(productDigits(1595)) | False |

def productDigits(n):

num\_str = str(n)

product\_even = 1

sum\_odd = 0

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

digit = int(digit)

if (i + 1) % 2 == 0:

product\_even \*= digit

else:

sum\_odd += digit

if sum\_odd == 0:

return "False"

elif product\_even % sum\_odd == 0:

return "True"

else:

return "False"

TestExpectedGot

print(productDigits(1256))

True

True

print(productDigits(1595))

False

False

**Ex. No. : 9.4 Date: 29/05/2024**

**Register No:231401037 Name: ILANKO M**

**Ugly number**

A number is considered to be ugly if its only prime factors are 2, 3 or 5.

[1, 2, 3, 4, 5, 6, 8, 9, 10, 12, 15, …] is the sequence of ugly numbers.

Task:

complete the function which takes a number n as input and checks if it's an ugly number.

return ugly if it is ugly, else return not ugly

Hint:

An ugly number U can be expressed as: U = 2^a \* 3^b \* 5^c, where a, b and c are nonnegative integers.

**Program:**

def checkUgly(n):

if n <= 0:

return "not ugly"

while n % 2 == 0:

n //= 2

while n % 3 == 0:

n //= 3

while n % 5 == 0:

n //= 5

if n == 1:

return "ugly"

else:

return "not ugly"

**For example:**



**Ex. No. : 9.5 Date: 29/05/2024**

**Register No:231401037 Name: ILANKO M**

**Christmas Discount**

An e-commerce company plans to give their customers a special discount for Christmas.

They are planning to offer a flat discount. The discount value is calculated as the sum of all

the prime digits in the total bill amount.

Write an algorithm to find the discount value for the given total bill amount.

Constraints

1 <= orderValue< 10e100000

**Input**

The input consists of an integer orderValue, representing the total bill amount.

**Output**

Print an integer representing the discount value for the given total bill amount.

**Example Input**

578

**Output**

12

**Program:**

def christmasDiscount(n):

def is\_prime(num):

if num < 2:

return False

for i in range(2, int(num \*\* 0.5) + 1):

if num % i == 0:

return False

return True

discount = 0

for digit in str(n):

digit\_int = int(digit)

if is\_prime(digit\_int):

discount += digit\_int

return discount

**For example:**

