**07-Functions**

**Ex. No. : 7.1 Date: 29/5/24**

**Register No.: 231801171 Name: Stergio Eugin**

## AbundantNumber

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

**InputFormat**:

Takeinputanintegerfromstdin

**OutputFormat:**

ReturnYesifgivennumberisAbundant.Otherwise,printNo

**Exampleinput:**

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**

Theproperdivisorsof13is:1,whosesumis1.Sincesumofproperdivisorsisnot greater than the given number, 13 is not an abundant number.

Forexample:

Test Result

print(abundant(12)) Yes

print(abundant(13)) No

# Program:

defabundant(n): l,s=[],0

foriinrange(1,int(n//2)+1): if(n%i==0):

l.append(i) for i in l:

s+=iif(s>n):

return("Yes") else:

return("No")

# Output:



**Ex. No. : 7.2 Date: 29/5/24**

**Register No.: 231801171 Name: Stergio Eugin**

## Automorphicnumberornot

An automorphic number is a number whose square ends with the number itself. For example,5isanautomorphicnumberbecause5\*5=25.Thelastdigitis5whichsame as the given number.

Ifthenumberisnotvalid,itshoulddisplay“Invalidinput”.

Ifitisanautomorphicnumberdisplay“Automorphic”elsedisplay“NotAutomorphic”.

InputFormat:

TakeaIntegerfromStdin Output Format:

PrintAutomorphicifgivennumberisAutomorphicnumber,otherwiseNot Automorphic

Exampleinput:5Output:AutomorphicExampleinput:25Output:Automorphic Example input: 7 Output: Not Automorphic

Forexample:

Test Result

print(automorphic(5)) Automorphic

# Program:

defautomorphic(n): a=str(n\*n) if(int(a[-1])==n):

return("Automorphic") else:

return("NotAutomorphic")

# Output:



**Ex. No. : 7.3 Date: 29/5/24**

**Register No.: 231801171 Name: Stergio Eugin**

## CheckProductofDigits

Write a code to check whether product of digits at even places isdivisible bysum of digits at odd place of a positive integer.

InputFormat:

Takeaninputintegerfromstdin. Output Format:

PrintTRUEorFALSE. Example Input:

1256

Output:

TRUE

ExampleInput: 1595

Output: FALSE

Forexample:

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

# Program:

defproductDigits(n):

a=str(n) s,p=0,1

foriinrange(0,len(a),2): s+=int(a[i])

foriinrange(1,len(a),2): p\*=int(a[i])

if(p%s==0): return("True")

else:

return("False")

# Output:



**Ex. No. : 7.4 Date: 29/5/24**

**Register No.: 231801171 Name: Stergio Eugin**

## ChristmasDiscount

Ane-commercecompanyplanstogivetheircustomersaspecialdiscountfor Christmas.

Theyareplanningtoofferaflatdiscount.The discountvalueiscalculatedasthe sum of all the prime digits in the total bill amount.

Writeanpythoncodetofindthediscountvalueforthegiventotalbillamount.

**Constraints**

1<=orderValue<10e100000 Input

Theinputconsists ofanintegerorderValue, representingthetotalbillamount. Output

Printanintegerrepresentingthe discountvalueforthegiventotalbillamount. Example Input

578

Output 12

**For example:**

|  |  |
| --- | --- |
| **Test** | **Result** |
| print(christmasDiscount(578)) | 12 |

# Program:

defchristmasDiscount(n): res=0

while n!=0: rem=n%10 flag=0

foriinrange(1,rem+1): if rem%i==0:

flag+=1 if flag==2:

res=res+rem n=n//10

returnres

# Output:



**Ex. No. : 7.5 Date: 29/5/24**

**Register No.: 231801171 Name: Stergio Eugin**

## CoinChange

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

Theonlyavailablecoinsareofvalues1,2,3,4 Input Format:

Integerinputfromstdin. Output Format:

returntheminimumnumberof coinsrequiredtomeetthegiventarget. Example Input:

16

Output:

4

Explanation:

Weneedonly4coinsofvalue4each Example Input:

25

Output:

7

Explanation:

Weneed6coinsof4value,and1coinof1value

# Program:

**defcoinChange(amount):**

**#Availablecoindenominations coins = [1, 2, 3, 4]**

**#Initializealisttostore theminimumnumberofcoinsforeachamount from 0 to the target amount**

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

**dp[0]=0#Basecase:0coinsneededtomakeamount0**

**#Iteratethroughallamountsfrom1tothetargetamount for i in range(1, amount + 1):**

**#Iteratethroughallavailablecoindenominations for coin in coins:**

**#Ifthecurrentcoindenominationislessthanorequaltothe current amount**

**ifcoin<= i:**

**#Updatedp[i]tobetheminimumbetweenitscurrentvalue and dp[i - coin] + 1**

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

**#Theresultisstoredatdp[amount] return dp[amount]**

**amount = int(input()) print(coinChange(amount))**

# Output:



**Ex. No. : 7.6 Date: 29/5/24**

**Register No.: 231801171 Name: Stergio Eugin**

## DifferenceSum

Givenanumberwithmaximumof100digitsasinput, findthedifferencebetween the sum of odd and even position digits.

InputFormat:

TakeanumberintheformofStringfromstdin. Output Format:

Printthedifferencebetweensumofevenand odddigits Example input:

1453

Output:

1

Explanation:

Here,sumofevendigitsis4+3=7 sum of odd digits is 1 + 5 = 6.

Differenceis1.

Notethatwearealwaystakingabsolutedifference

# Program:

**defdifferenceSum(n): a=[]**

**b=[]**

**k=str(n)**

**foriinrange(len(k)): if int(i)%2==0:**

**a.append(int(k[i])) else:**

**b.append(int(k[i])) s=sum(b)**

**r=sum(a) j=s-r return j**

# Output:



**Ex. No. : 7.7 Date: 29/5/24**

**Register No.: 231801171 Name: Stergio Eugin**

## Ugly number

Anumberisconsideredtobeuglyifitsonlyprimefactorsare2, 3or5. [1, 2, 3, 4, 5, 6, 8, 9, 10, 12, 15, …] is the sequence of ugly numbers.

Task:

completethefunctionwhichtakesanumbernasinputandchecksifit'sanugly number. return ugly if it is ugly, else return not ugly

Hint:

An uglynumber Ucanbeexpressedas: U=2^a\*3^b\*5^c,wherea, bandcare nonnegative integers.

**For example:**

|  |  |
| --- | --- |
| **Test** | **Result** |
| print(checkUgly(6)) | ugly |
| print(checkUgly(21)) | notugly |

# Program:

**defcheckUgly(n):**

**foriinrange(n):**

**forjinrange(n):**

**forkinrange(n):**

**if(n==(2\*\*i)+(3\*\*j)+(5\*\*k)): return("ugly")**

**return("notugly")**

Output:

