.

**Department of Computer Science and Business Systems** | **Rajalakshmi Engineering College**

74



04 - Iteration Control Structures

## For example:



**Input Result**

20

1 2 4 5 10 20

.

**Department of Computer Science and Business Systems** | **Rajalakshmi Engineering College**

75



**Ex. No. : 4.1 Date: 26.03.2024**

.

**Department of Computer Science and Business Systems** | **Rajalakshmi Engineering College**

76



**Register No.:231401038 Name: INIYA S**

# Factors of a number

Determine the factors of a number (i.e., all positive integer values that evenly divide into a number).

CODE:

a=int(input())

for i in range(1,21): if a%i==0:

print(i,end=" ")

## For example:

.

**Department of Computer Science and Business Systems** | **Rajalakshmi Engineering College**

77



|  |  |
| --- | --- |
| **Input** | **Result** |
| 292 | 1 |
| 1015 | 2 |
| 108 | 3 |
| 22 | 0 |

**Ex. No. : 4.2 Date: 26.03.2024**



**Register No.:231401038 Name: INIYA S**

# Non Repeated Digit Count

Write a program to find the count of non-repeated digits in a given number N. The number will be passed to the program as an input of type int.

Assumption: The input number will be a positive integer number >= 1 and <= 25000. Some examples are as below.

If the given number is 292, the program should return 1 because there is only 1 non-- repeated digit '9' in this number

If the given number is 1015, the program should return 2 because there are 2 non- repeated digits in this number, '0', and '5'.

If the given number is 108, the program should return 3 because there are 3 non-- repeated digits in this number, '1', '0', and '8'.

If the given number is 22, the function should return 0 because there are NO non-- repeated digits in this number.

**CODE:**

def digits(n): count=0

for digit in range(10):

if str(n).count(str(digit))==1: count+=1

return count n=int(input()) print(digits(n))

.

**Department of Computer Science and Business Systems** | **Rajalakshmi Engineering College**

78

Example1: if the given number N is 7, the method must return 2 Example2: if the given number N is 10, the method must return 1

.

**Department of Computer Science and Business Systems** | **Rajalakshmi Engineering College**

79



## For example:

|  |  |
| --- | --- |
| **Input** | **Result** |
| 7 | 2 |
| 10 | 1 |

**Ex. No. : 4.3 Date: 26.03.2024**

.

**Department of Computer Science and Business Systems** | **Rajalakshmi Engineering College**

80



**Register No.:231401038 Name: INIYA S**

# Prime Checking

Write a program that finds whether the given number N is Prime or not. If the number is prime, the program should return 2 else it must return 1.

Assumption: 2 <= N <=5000, where N is the given number. CODE:

def prime(N): if N<2:

return 1

for i in range(2,int(N\*\*0.5)+1): if N%i==0:

return 1

return 2 N=int(input()) print(prime(N))

Input Format:

.

**Department of Computer Science and Business Systems** | **Rajalakshmi Engineering College**

81



Integer input from stdin. Output Format:

Perfect square greater than N. Example Input:

10

Output: 16

**Ex. No. : 4.4 Date: 26.03.2024**

.

**Department of Computer Science and Business Systems** | **Rajalakshmi Engineering College**

82



**Register No.:231401038 Name: INIYA S**

# Next Perfect Square

Given a number N, find the next perfect square greater than N.

CODE:

from math import sqrt n=int(input())

while int(sqrt(n))!=sqrt(n): n=n+1

print(n)

NOTE: Fibonacci series looks like –

.

**Department of Computer Science and Business Systems** | **Rajalakshmi Engineering College**

83



0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, . . . and so on.

i.e. Fibonacci series starts with 0 and 1, and continues generating the next number as the sum of the previous two numbers.

* first Fibonacci number is 0,
* second Fibonacci number is 1,
* third Fibonacci number is 1,
* fourth Fibonacci number is 2,
* fifth Fibonacci number is 3,
* sixth Fibonacci number is 5,
* seventh Fibonacci number is 8, and so on.

## For example: Input:

**7**

## Output 8

**Ex. No. : 4.5 Date: 26.03.2024**

.

**Department of Computer Science and Business Systems** | **Rajalakshmi Engineering College**

84



**Register No.:231401038 Name: INIYA S**

# Nth Fibonacci

Write a program to return the nth number in the fibonacci series. The value of N will be passed to the program as input.

CODE:

def fi(n): if n<0:

return "" elif n==1:

return 0 elif n==2:

return 1 else:

return fi(n-1)+fi(n-2) n=int(input())

print(fi(n))

Input Format:

.

**Department of Computer Science and Business Systems** | **Rajalakshmi Engineering College**

85



Single Integer Input from stdin. Output Format:

Yes or No. Example Input: 175

Output:

Yes Explanation

1^1 + 7^2 +5^3 = 175

Example Input: 123

Output:

No

## For example: InputResult

175 Yes

123 No

**Ex. No. : 4.6 Date: 26.03.2024**

.

**Department of Computer Science and Business Systems** | **Rajalakshmi Engineering College**

86



**Register No.:231401038 Name: INIYA S**

# Disarium Number

A Number is said to be Disarium number when the sum of its digit raised to the power of their respective positions becomes equal to the number itself. Write a program to print number is Disarium or not.

CODE:

def dis(num): num\_str=str(num) sum=0

for i in range(len(num\_str)): sum+=int(num\_str[i])\*\*(i+1)

if sum==num: return "Yes"

else:

return "No" num=int(input()) print(dis(num))

Sample Test Cases Test Case 1

.

**Department of Computer Science and Business Systems** | **Rajalakshmi Engineering College**

87



Input 4

Output 1234

Explanation:

as input is 4, have to take 4 terms. 1 + 11 + 111 + 1111

Test Case 2 Input

6

Output 123456

## For example:



**Input Result**

3

123

**Ex. No. : 4.7 Date: 26.03.2024**

.

**Department of Computer Science and Business Systems** | **Rajalakshmi Engineering College**

88



**Register No.:231401038 Name: INIYA S**

# Sum of Series

Write a program to find the sum of the series 1 +11 + 111 + 1111 + . . . + n terms (n will be given as input from the user and sum will be the output)

CODE:

n=int(input()) sum=0 term=1

for i in range(n): sum+=term term=term\*10+1

print(sum)

## For example:

.

**Department of Computer Science and Business Systems** | **Rajalakshmi Engineering College**

89



|  |  |
| --- | --- |
| **Input** | **Result** |
| 292 | 2 |
| 1015 | 3 |

**Ex. No. : 4.8 Date: 26.03.2024**

.

**Department of Computer Science and Business Systems** | **Rajalakshmi Engineering College**

90



**Register No.:231401038 Name: INIYA S**

# Unique Digit Count

Write a program to find the count of unique digits in a given number N. The number will be passed to the program as an input of type int.

Assumption: The input number will be a positive integer number >= 1 and <= 25000. For e.g.

If the given number is 292, the program should return 2 because there are only 2 unique digits '2' and '9' in this number

If the given number is 1015, the program should return 3 because there are 3 unique digits in this number, '1', '0', and '5'.

CODE:

def digits(N):

return len(set(str(N))) N=int(input()) print(digits(N))

Input Format:

.

**Department of Computer Science and Business Systems** | **Rajalakshmi Engineering College**

91



Single Integer input. Output Format:

Output displays Yes if condition satisfies else prints No. Example Input:

14

Output:

Yes

Example Input: 13

Output: No

**Ex. No. : 4.9 Date: 26.03.2024**

.

**Department of Computer Science and Business Systems** | **Rajalakshmi Engineering College**

92



**Register No.:231401038 Name: INIYA S**

# Product of single digit

Given a positive integer N, check whether it can be represented as a product of single digit numbers.

CODE:

a=int(input())

if a%2==0 or a%3==0 or a%5==0 or a%7==0 or a%9==0: print("Yes")

else:

print("No")

Input Format: Single integer input. Output Format:

.

**Department of Computer Science and Business Systems** | **Rajalakshmi Engineering College**

93



Yes or No. Example Input: 24

Output:

Yes

Example Input: 26

Output:

No

## For example:



**Input Result**

24

Yes

**Ex. No. : 4.10 Date: 26.03.2024**

.

**Department of Computer Science and Business Systems** | **Rajalakshmi Engineering College**

94



**Register No.:231401038 Name: INIYA S**

# Perfect Square After adding One

Given an integer N, check whether N the given number can be made a perfect square after adding 1 to it.

CODE:

def sq(n): if(n)%4==0:

return "Yes" else:

return "No" n=int(input()) print(sq(n))

.

**Department of Computer Science and Business Systems** | **Rajalakshmi Engineering College**

95

