

DATA SCIENCE LAB

CYCLE-1

1. Program to Print all non-Prime Numbers in an Interval.

CODE:

```
print("Name:Gopika Unnikrishnan\nRoll No:22MCA030\nCourse Name:DATA SCIENCE LAB\nCourse Code:20MCA241\nDate:25/09/2023")
x=int(input("Enter lower digit:"))
y=int(input("Enter upper digit:"))
print("lower=",x)
print("upper",y)
print("The prime numbers in between the range ",x,"to",y)
for n in range(x, y+1):
    if(n > 1):
        for i in range(2,n):
            if(n % i) == 0:
                print(n)
                break
```

OUTPUT:

```
Name:Gopika Unnikrishnan
Roll No:22MCA030
Course Name:DATA SCIENCE LAB
Course Code:20MCA241
Date:25/09/2023
Enter lower digit:10
Enter upper digit:20
lower= 10
upper 20
The prime numbers in between the range  10 to 20
10
12
14
15
16
18
20
```

2. Program to print the first N Fibonacci numbers.

CODE:

```
print("Name:Gopika Unnikrishnan\nRoll No:22MCA030\nCourse Name:DATA SCIENCE  
LAB\nCourse Code:20MCA241")  
n = int(input("Number of terms:"))  
n1 = 0  
n2 = 1  
count = 0  
if n <= 0:  
    print("Please enter a Postive Integer")  
elif n==1:  
    print("Fibonacci series upto",n,":")  
    print(n1)  
else:  
    print("Fibonacci Series:")  
    while count < n:  
        print(n1)  
        nth=n1+n2  
        n1 = n2  
        n2 = nth  
        count += 1
```

OUTPUT:

```
Name:Gopika Unnikrishnan  
Roll No:22MCA030  
Course Name:DATA SCIENCE LAB  
Course Code:20MCA241  
Date:25/09/2023  
Number of terms:13  
Fibonacci Series:  
0  
1  
1  
2  
3  
5  
8  
13  
21  
34  
55  
89  
144
```

3. Given sides of a triangle, write a program to check whether given triangle is an isosceles, equilateral or scalene.

CODE:

```
print("Name:Gopika Unnikrishnan\nRoll No:22MCA030\nCourse Name:DATA SCIENCE  
LAB\nCourse Code:20MCA241\nDate:25/09/2023")  
x=int(input("Enter lower digit:"))  
y=int(input("Enter upper digit:"))  
print("lower=",x)  
print("upper",y)  
print("The prime numbers in between the range ",x,"to",y)  
for n in range(x, y+1):  
    if(n > 1):  
        for i in range(2,n):  
            if(n % i) == 0:  
                print(n)  
                break
```

OUTPUT:

```
Name:Gopika Unnikrishnan  
Roll No:22MCA030  
Course Name:DATA SCIENCE LAB  
Course Code:20MCA241  
Date:25/09/2023  
Enter the side(x):15  
Enter the side(y):15  
Enter the side(z):15  
Triangle is Equilateral
```

4. Program to check whether given pair of number is coprime

CODE:

```
print("Name:Gopika Unnikrishnan\nRoll No:22MCA030\nCourse Name:DATA SCIENCE  
LAB\nCourse Code:20MCA241\nDate:25/09/2023")  
def gcd(x,y):  
    if(x==0 or y==0):  
        return 0  
    if(x==y):  
        return x  
    if(x>y):  
        return gcd(x-y,y)  
    return gcd(x,y-x)  
def coprime(x,y):  
    if(gcd(x,y)==1):  
        print("The Numbers", (x,y), "are Coprime")  
    else:  
        print("The Numbers", (x, y), "are not Coprime")  
x=int(input("Enter the number(x):"))  
y=int(input("Enter the number(y):"))  
coprime(x,y)
```

OUTPUT:

```
Name:Gopika Unnikrishnan  
Roll No:22MCA030  
Course Name:DATA SCIENCE LAB  
Course Code:20MCA241  
Date:25/09/2023  
Enter the number(x):5  
Enter the number(y):6  
The Numbers (5, 6) are Coprime
```

5. Program to find the roots of a quadratic equation(rounded to 2 decimal places)

CODE:

```
print("Name:Gopika Unnikrishnan\nRoll No:22MCA030\nCourse Name:DATA SCIENCE LAB\nCourse Code:20MCA241\nDate:25/09/2023")
print("Equation : ax^2 + bx + c :")
a=int(input("Enter a :"))
b=int(input("Enter b :"))
c=int(input("Enter c :"))
d=b*b-4*a*c
if(d<0):
    print("The roots are imaginary")
else:
    r1=(-b+d)/2*a
    r2=(-b-d)/2*a
    print("The first root: ",round(r1,2))
    print("The second root: ",round(r2,2))
```

OUTPUT:

```
Name:Gopika Unnikrishnan
Roll No:22MCA030
Course Name:DATA SCIENCE LAB
Course Code:20MCA241
Date:25/09/2023
Equation : ax^2 + bx + c :
Enter a :4
Enter b :-5
Enter c :-12
The first root:  444.0
The second root:  444.0
```

6. Program to check whether a given number is perfect number or not(sum of factors =number)

CODE:

```
print("Name:Gopika Unnikrishnan\nRoll No:22MCA030\nCourse Name:DATA SCIENCE LAB\nCourse Code:20MCA241\nDate:25/09/2023")
n = int(input("Enter the number="))
sum=0
for i in range(1,n):
    if(n % i == 0):
        sum=sum+i
if(sum==n):
    print("The Number",n,"is a Perfect Number")
else:
    print("The Number", n, "is not Perfect Number")
```

OUTPUT:

```
Name:Gopika Unnikrishnan
Roll No:22MCA030
Course Name:DATA SCIENCE LAB
Course Code:20MCA241
Date:25/09/2023
Enter the number=28
The Number 28 is a Perfect Number
```

7. Program to display amstrong numbers upto 1000.

CODE:

```
print("Name:Gopika Unnikrishnan\nRoll No:22MCA030\nCourse Name:DATA SCIENCE  
LAB\nCourse Code:20MCA241\nDate:25/09/2023")  
lower = int(input("Enter the lower range:"))  
upper = int(input("Enter the upper range:"))  
  
for num in range(lower, upper + 1):  
    order = len(str(num))  
    sum = 0  
    temp = num  
    while temp > 0:  
        digit = temp % 10  
        sum += digit ** order  
        temp //= 10  
    if num == sum:  
        print(num)
```

OUTPUT:

```
Name:Gopika Unnikrishnan  
Roll No:22MCA030  
Course Name:DATA SCIENCE LAB  
Course Code:20MCA241  
Date:25/09/2023  
Enter the lower range:0  
Enter the upper range:1000  
0  
1  
2  
3  
4  
5  
6  
7  
8  
9  
153  
370  
371  
407
```

8. Store and display the days of a week as a List, Tuple, Dictionary, Set. Also demonstrate different ways to store values in each of them. Display its type also.

CODE:

```
print("Name:Gopika Unnikrishnan\nRoll No:22MCA030\nCourse Name:DATA SCIENCE LAB\nCourse Code:20MCA241\nDate:26/09/2023")
week_list=["Monday","Tuesday","Wednesday","Thursday","Friday","Saturday","Sunday"]
print("List of days(List): ",week_list)
print("Type of list: ",type(week_list))
week_tuple=("Monday","Tuesday","Wednesday","Thursday","Friday","Saturday","Sunday")
print("\nlist of days (Tuple): ",week_tuple)
print("Type of Tuple: ",type(week_tuple))
week_dict={1:"Monday",2:"Tuesday",3:"Wednesday",4:"Thursday",5:"Friday",6:"Saturday",7:"Sunday"}
print("\nList of Days(Dictionary): ",week_dict)
print("Type of Dictionary: ",type(week_dict))
week_set={"Monday","Tuesday","Wednesday","Thursday","Friday","Saturday","Sunday"}
print("\nList of days(set): ",week_set)
print("Type of Set: ",type(week_set))
```

OUTPUT:

```
Name:Gopika Unnikrishnan
Roll No:22MCA030
Course Name:DATA SCIENCE LAB
Course Code:20MCA241
Date:26/09/2023
List of days(List): ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']
Type of list: <class 'list'>

list of days (Tuple): ('Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday')
Type of Tuple: <class 'tuple'>

List of Days(Dictionary): {1: 'Monday', 2: 'Tuesday', 3: 'Wednesday', 4: 'Thursday', 5: 'Friday', 6: 'Saturday', 7: 'Sunday'}
Type of Dictionary: <class 'dict'>

List of days(set): {'Thursday', 'Friday', 'Saturday', 'Sunday', 'Wednesday', 'Monday', 'Tuesday'}
Type of Set: <class 'set'>
```


9. Write a program to add elements of given 2 lists.

CODE:

```
print("Name:Gopika Unnikrishnan\nRoll No:22MCA030\nCourse Name:DATA SCIENCE  
LAB\nCourse Code:20MCA241\nDate:26/09/2023")  
list1 = [ ]  
n = int(input("Enter number of elements : "))  
for i in range(0, n):  
    num1 = int(input())  
    list1.append(num1)  
print("The List-1:",list1)  
list2 = [ ]  
n = int(input("Enter number of elements : "))  
for i in range(0, n):  
    num2 = int(input())  
    list2.append(num2)  
print( "The List-2:",list2)  
sum_list = [ ]  
for i in range(0, len(list1)):  
    sum_list.append(list1[i] + list2[i])  
print("Resultant list is :"+ str(sum_list))
```

OUTPUT:

```
Name:Gopika Unnikrishnan  
Roll No:22MCA030  
Course Name:DATA SCIENCE LAB  
Course Code:20MCA241  
Date:26/09/2023  
Enter number of elements : 3  
5  
10  
15  
The List-1: [5, 10, 15]  
Enter number of elements : 3  
10  
20  
30  
The List-2: [10, 20, 30]  
Resultant list is :[15, 30, 45]
```

10. Write a program to find the sum of 2 matrices using nested List.

CODE:

```
def add_matrices(matrix1, matrix2):

    if len(matrix1) != len(matrix2) or len(matrix1[0]) != len(matrix2[0]):
        return None
    result = [[0 for _ in range(len(matrix1[0]))] for _ in range(len(matrix1))]

    for i in range(len(matrix1)):
        for j in range(len(matrix1[0])):
            result[i][j] = matrix1[i][j] + matrix2[i][j]

    return result

try:
    print("Name:Gopika Unnikrishnan\nRoll No:22MCA030\nCourse Name:DATA SCIENCE LAB\nCourse Code:20MCA241\nDate:26/09/2023")
    rows = int(input("Enter the number of rows: "))
    cols = int(input("Enter the number of columns: "))

    print("Enter elements of the first matrix:")
    matrix1 = []
    for i in range(rows):
        row = input(f"Enter elements of row {i + 1} separated by spaces: ").split()
        matrix1.append([int(x) for x in row])

    print("Enter elements of the second matrix:")
    matrix2 = []
    for i in range(rows):
        row = input(f"Enter elements of row {i + 1} separated by spaces: ").split()
        matrix2.append([int(x) for x in row])
    result = add_matrices(matrix1, matrix2)

    if result is None:
        print("Matrix dimensions are not compatible for addition.")
    else:
        print("Sum of matrices:")
        for row in result:
            print(" ".join(map(str, row)))
```

```
except ValueError:  
    print("Invalid input. Please enter valid numbers.")
```

OUTPUT:

```
Name:Gopika Unnikrishnan  
Roll No:22MCA030  
Course Name:DATA SCIENCE LAB  
Course Code:20MCA241  
Date:26/09/2023  
Enter the number of rows: 2  
Enter the number of columns: 2  
Enter elements of the first matrix:  
Enter elements of row 1 separated by spaces: 1 3  
Enter elements of row 2 separated by spaces: 5 8  
Enter elements of the second matrix:  
Enter elements of row 1 separated by spaces: 7 9  
Enter elements of row 2 separated by spaces: 12 15  
Sum of matrices:  
8 12  
17 23
```

11. Write a program to perform bubble sort on a given set of elements.

CODE:

```
def bubble_sort(arr):
    n = len(arr)
    for i in range(n):
        swapped = False
        for j in range(0, n - i - 1):
            if arr[j] > arr[j + 1]:
                arr[j], arr[j + 1] = arr[j + 1], arr[j]
                swapped = True
        if not swapped:
            break
try:
    print(
        "Name:Gopika Unnikrishnan\nRoll No:22MCA030\nCourse Name:DATA SCIENCE
LAB\nCourse Code:20MCA241\nDate:03/10/2023")
    elements = input("Enter elements separated by spaces: ").split()
    elements = [int(x) for x in elements]
    bubble_sort(elements)
    print("Sorted elements:")
    print(elements)
except ValueError:
    print("Invalid input. Please enter valid numbers separated by spaces.")
```

OUTPUT:

```
Name:Gopika Unnikrishnan
Roll No:22MCA030
Course Name:DATA SCIENCE LAB
Course Code:20MCA241
Date:03/10/2023
Enter elements separated by spaces: 12 14 17 09 45
Sorted elements:
[9, 12, 14, 17, 45]
```

12. Program to find the count of each vowel in a string(use dictionary)

CODE:

```
vowels='aeiou'
print("Name:Gopika Unnikrishnan\nRoll No:22MCA030\nCourse Name:DATA SCIENCE
LAB\nCourse Code:20MCA241\nDate:03/10/2023")
a=input("Enter the string:")
a=a.casefold()
count={}.fromkeys(vowels,0)
for char in a:
    if char in count:
        count[char] +=1
print(count)
```

OUTPUT:

```
Name:Gopika Unnikrishnan
Roll No:22MCA030
Course Name:DATA SCIENCE LAB
Course Code:20MCA241
Date:03/10/2023
Enter the string:COMPUTER
{'a': 0, 'e': 1, 'i': 0, 'o': 1, 'u': 1}
```

13. Write a Python program that accept a positive number and subtract from this number the sum of its digits and so on. Continues this operation until the number is positive.

(eg: $256 \rightarrow 2+5+6=13, 256-13=243, 243-9=232 \dots$)

CODE:

```
def sum_of_digits(number):
    return sum(int(digit) for digit in str(number))
def main():
    try:
        print("Name:Gopika Unnikrishnan\nRoll No:22MCA030\nCourse Name:DATA SCIENCE LAB\nCourse Code:20MCA241\nDate:03/10/2023")
        number = int(input("Enter a positive number: "))
        if number <= 0:
            print("Please enter a positive number.")
            return
        while number > 0:
            digit_sum = sum_of_digits(number)
            print(f"{number} - {digit_sum} = {number - digit_sum}")
            number = number - digit_sum
        except ValueError:
            print("Invalid input. Please enter a valid positive number.")
if __name__ == "__main__":
    main()
```

OUTPUT:

```
Name:Gopika Unnikrishnan
Roll No:22MCA030
Course Name:DATA SCIENCE LAB
Course Code:20MCA241
Date:03/10/2023
Enter a positive number: 102
102 - 3 = 99
99 - 18 = 81
81 - 9 = 72
72 - 9 = 63
63 - 9 = 54
54 - 9 = 45
45 - 9 = 36
36 - 9 = 27
27 - 9 = 18
18 - 9 = 9
9 - 9 = 0
```

14. Write a Python program that accepts a 10 digit mobile number, and find the digits which are absent in a given mobile number.

CODE:

```
def find_absent_digits(mobile_number):
    all_digits = set("0123456789")
    mobile_digits = set(mobile_number)
    absent_digits = all_digits - mobile_digits
    return sorted(list(absent_digits))

try:
    print("Name:Gopika Unnikrishnan\nRoll No:22MCA030\nCourse Name:DATA SCIENCE LAB\nCourse Code:20MCA241\nDate:03/10/2023")
    mobile_number = input("Enter a 10-digit mobile number: ")
    if len(mobile_number) == 10 and mobile_number.isdigit():
        absent_digits = find_absent_digits(mobile_number)
        if absent_digits:
            print("Absent digits in the mobile number:", ', '.join(absent_digits))
        else:
            print("The mobile number contains all digits from 0 to 9.")
    else:
        print("Invalid input. Please enter a valid 10-digit mobile number.")
except ValueError:
    print("Invalid input. Please enter a valid 10-digit mobile number.")
```

OUTPUT:

```
Name:Gopika Unnikrishnan
Roll No:22MCA030
Course Name:DATA SCIENCE LAB
Course Code:20MCA241
Date:03/10/2023
Enter a 10-digit mobile number: 9188536782
Absent digits in the mobile number: 0, 4
```