

LAB CYCLE 1

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

Code:

```
print("\nNafeesath Neema\nRoll n0:42\nbatch no:MCA-2022-24")
lower = int(input("enter the lower limit"))
upper = int(input("enter the upper limit"))
print("prime numbers between",lower,"and",upper,"are")
for num in range(lower,upper + 1):
    if num > 1:
        for i in range(2,num):
            if (num % i) == 0:
                print(num)
                break
```

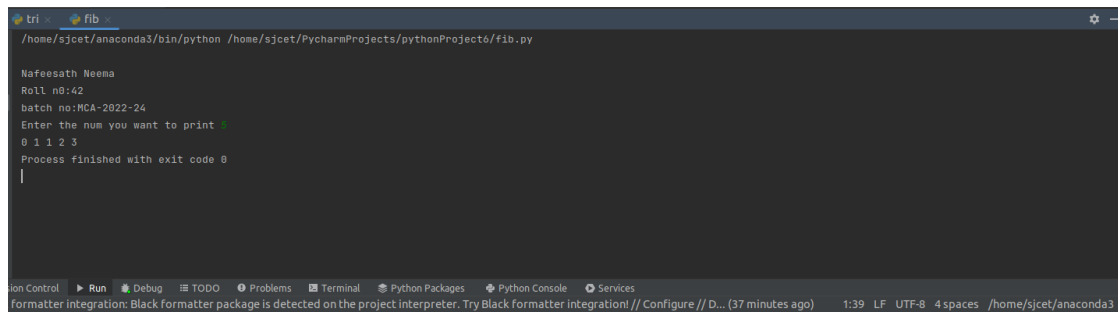
```
Nafeesath Neema
Roll n0:42
batch no:MCA-2022-24
enter the lower limit2
enter the upper limit10
prime numbers between 2 and 10 are
4
6
8
9
10
```

2. Program to print the first N Fibonacci numbers.

Code:

```
print("\nNafeesath Neema\nRoll n0:42\nbatch no:MCA-2022-24")
n = int(input("Enter the num you want to print"))
a = 0
b = 1
for i in range(0,n):
    print(a,end=" ")
    c = (a + b)
    a = b
    b = c
```

Output:



```
trf x fib x
/home/sjcet/anaconda3/bin/python /home/sjcet/PycharmProjects/pythonProject6/fib.py

Nafeesath Neema
Roll n0:42
batch no:MCA-2022-24
Enter the num you want to print
0 1 1 2 3
Process finished with exit code 0
|

Run Debug TODO Problems Terminal Python Packages Python Console Services
formatter integration: Black formatter package is detected on the project interpreter. Try Black formatter integration! // Configure // D... (37 minutes ago) 1:39 LF UTF-8 4 spaces /home/sjcet/anaconda3
```

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

Code:

```
print("\nNafeesath Neema\nRoll n0:42\nbatch no:MCA-2022-24")
print("input len of the triangle sides:")
x = int(input("x: "))
y = int(input("y: "))
z = int(input("z: "))
if x == y == z:
    print("Equilateral triangle")
elif x == y or y == z or z == x:
    print("isoscelec triangle")
else:
    print("SCALENE TRIANGLE")
```

Output:

```
Nafeesath Neema
Roll n0:42
batch no:MCA-2022-24
input len of the triangle sides:
x: 12
y: 12
z: 12
Equilateral triangle

Process finished with exit code 0
|
```

```
Nafeesath Neema
Roll n0:42
batch no:MCA-2022-24
input len of the triangle sides:
x: 12
y: 12
z: 4
isoscelec triangle

Process finished with exit code 0
|
```

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

Code:

```
print("\nNafeesath Neema\nRoll n0:42\nbatch no:MCA-2022-24")

def are_coprime(a,b):
    hcf = 1
    for i in range(1,a+1):
        if a % i ==0 and b % i ==0:
            hcf = i
    return hcf == 1

first = int(input("enter first number"))
secnd = int(input("enter the second number"))
if are_coprime(first, secnd):
    print('%d and %d are co-PRIME'%(first, secnd))
else:
    print('%d and %d are not co-PRIME' %(first, secnd))
```

Output:

```
Nafeesath Neema
Roll n0:42
batch no:MCA-2022-24
enter first number2
enter the second number7
2 and 7 are co-PRIME
```

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

```
print("\nNafeesath Neema\nRoll n0:42\nbatch no:MCA-2022-24")
print("Equation: ax^2 + bx + c ")
a = int(input("Enter a:"))
b = int(input("Enter b: "))
c = int(input("enter c: "))
d = b ** 2 - 4 * a * c
d1 = d ** 0.5
if d < 0:
    print("The roots are imaginary")
else:
    r1 = (- b + d1)/2 * a
    r2 = (-b - d1)/2 * a
    print("The first root: ", round(r1, 2))
    print("The second root: ", round(r2, 2))
```

Output:

```
Nafeesath Neema
Roll n0:42
batch no:MCA-2022-24
Equation: ax^2 + bx + c
Enter a:2
Enter b: -10
enter c: 5
The first root:  17.75
The second root:  2.25
```

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

```
um = int(input("enter the number:"))
sum_v = 0
for i in range(1,num):
    if num % i == 0:
        sum_v = sum_v + i
if sum_v == num:
    print("The entered num is perfect num")
else:
    print("The entered num is not perfect number")
```

Output:

```
Nafeesath Neema
Roll n0:42
batch no:MCA-2022-24
enter the number:25
The entered num is not perfect number
```

7. Program to display armstrong numbers upto 1000

Code:

```
print("\nNafeesath Neema\nRoll n0:42\nbatch no:MCA-2022-24")
print("Armstrong num up to 1000 are:")
for num in range(1, 1001):
    num_digits = len(str(num))
    sum = 0
    temp = num
    while temp > 0:
        digit = temp % 10
        sum += digit ** num_digits
        temp //= 10
    if num == sum:
        print(num)
```

Output:

```
Nafeesath Neema
Roll n0:42
batch no:MCA-2022-24
Armstrong num up to 1000 are:
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.

```
print("\nNafeesath Neema\nRoll n0:42\nbatch no:MCA-2022-24")
days_list =
["Monday","Tuesday","Wednesday","Thursday","friday","Saturday","Sunday"]
print("List:", days_list)
print("Types of List", type(days_list))

days_tuple = ("Monday","Tuesday","wednesday","thursday","friday")
print("Tuples:", days_tuple)
print("Type of Tuple",type(days_tuple))

days_dict = {1: "Monday", 2: "Tuesday", 3: "Wednesday", 4: "Thursday", 5:
"Friday", 6: "Saturday", 7: "Sunday"}
print("Dictionary:", days_dict)
print("Type of Dictionary:", type(days_dict))

days_set =
{"Monday","Tuesday","Wednesday","Thursday","Friday","Saturday","Sunday"}
print("Set:", days_set)
print("Type of Set:", type(days_set))
```

```
Nafeesath Neema
Roll n0:42
batch no:MCA-2022-24
List: ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'friday', 'Saturday', 'Sunday']
Types of List <class 'list'>
Tuples: ('Monday', 'Tuesday', 'wednesday', 'thursday', 'friday')
Type of Tuple <class 'tuple'>
Dictionary: {1: 'Monday', 2: 'Tuesday', 3: 'Wednesday', 4: 'Thursday', 5: 'Friday', 6: 'Saturday', 7: 'Sunday'}
Type of Dictionary: <class 'dict'>
Set: {'Monday', 'Wednesday', 'Saturday', 'Thursday', 'Sunday', 'Friday', 'Tuesday'}
Type of Set: <class 'set'>
```


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

```
print("\nNafeesath Neema\nRoll n0:42\nbatch no:MCA-2022-24")
input_str1 = input("Enter element list1 separated by spaces")
list1 = input_str1.split()
input_str2 = input("Enter the elements list2 separated by spaces:")
list2 = input_str2.split()
if len(list1) != len(list2):
    print("Error:Lists must have the same length for element-wise addition")
else:
    list1 = [int(item) for item in list1]
    list2 = [int(item) for item in list2]
    result = [a + b for a, b in zip(list1, list2) ]
    print("Element-wise sum of the two lists", result)
```

```
Nafeesath Neema
Roll n0:42
batch no:MCA-2022-24
Enter element list1 separated by spaces:1 2 3
Enter the elements list2 separated by spaces:4 5
Error:Lists must have the same length for element-wise addition
```

```
Nafeesath Neema
Roll n0:42
batch no:MCA-2022-24
Enter element list1 separated by spaces:1 2 3 4
Enter the elements list2 separated by spaces:5 6 4 3
Element-wise sum of the two lists [6, 8, 7, 7]
```

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

```
print("\nNafeesath Neema\nRoll no:42\nbatch no:MCA-2022-24")
n = int(input("enter the order"))
m1 = []
print("enter the elements:")
for i in range(n):
    m1.append([])
    for j in range(n):
        num = int(input())
        m1[i].append(num)
m2 = []
print("enter the elements")
for i in range(n):
    m2.append([])
    for j in range(n):
        num = int(input())
        m2[i].append(num)
print("first matrix is :")
for i in range(n):
    for j in range(n):
        print(m1[i][j], end=" ")
    print()
print("second matrix is :")
for i in range(n):
    for j in range(n):
        print(m2[i][j], end=" ")
    print()
m3 = []
for i in range(n):
    m3.append([])
    for j in range(n):
        m3[i].append(m1[i][j] + m2[i][j])
print("resultant matrix is :")
for i in range(n):
    for j in range(n):
        print(m3[i][j], end=" ")
    print()
```

```
Nafeesath Neema
Roll n0:42
batch no:MCA-2022-24
enter the order:2
enter the elements:
1
2
4
5
enter the elements:
5
67
6
5
first matrix is:
1 2
4 5
second matrix is:
5 67
6 5
Resultent matrix is:
6 69
10 10
```

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

```
print("\nNafeesath Neema\nRoll n0:42\nbatch no:MCA-2022-24")
def bubble_sort(arr):
    n = len(arr)
    for i in range(n - 1):
        swapped = False
        for j in range(n - 1 - i):
            if arr[j] > arr[j + 1]:
                arr[j], arr[j + 1] = arr[j + 1], arr[j]
                swapped = True
        if not swapped:
            break
input_str = input("Enter elements : ")
elements = [int(x) for x in input_str.split()]
print("Original List:", elements)
bubble_sort(elements)
print("Sorted List:", elements)
```

```
Nafeesath Neema
Roll n0:42
batch no:MCA-2022-24
Enter elements : 12 34 13 45
Original List: [12, 34, 13, 45]
Sorted List: [12, 13, 34, 45]
```

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

```
print("\nNafeesath Neema\nRoll n0:42\nbatch no:MCA-2022-24")
def count_vowels(input_string):
    vowel_counts = {'a': 0, 'e': 0, 'i': 0, 'o': 0, 'u': 0}
    input_string = input_string.lower()
    for char in input_string:
        if char in vowel_counts:
            vowel_counts[char] += 1
    return vowel_counts
input_string = input("Enter a string: ")
vowel_count = count_vowels(input_string)
for vowel, count in vowel_count.items():
    print(f"{vowel}: {count}")
```

```
Roll n0:42
batch no:MCA-2022-24
Enter a string: neema
a: 1
e: 2
i: 0
o: 0
u: 0
```

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: $256 - 2 + 5 + 6 = 13$
 $256 - 13 = 243$
 $243 - 9 = 232$

```
print("\nNafeesath Neema\nRoll n0:42\nbatch no:MCA-2022-24")
def sum_of_digits(n):
    digit_sum = 0
    while n > 0:
        digit_sum += n % 10
        n //= 10
    return digit_sum
num = int(input("Enter a positive number: "))
while num > 0:
    digit_sum = sum_of_digits(num)
    num -= digit_sum
    print(f"{num + digit_sum} - {digit_sum} = {num}")
    print("The number is now positive or zero.")
```

```
Nafeesath Neema
Roll n0:42
batch no:MCA-2022-24
Enter a positive number: 14
14 - 5 = 9
The number is now positive or zero.
9 - 9 = 0
The number is now positive or zero.
```

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

```
print("\nNafeesath Neema\nRoll n0:42\nbatch no:MCA-2022-24")
def find_absent_digits(mobile_number):
    all_digits = set("0123456789")
    number_digits = set(mobile_number)
    absent_digits = all_digits - number_digits
    return absent_digits
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:", ', '.join(absent_digits))
    else:
        print("All digits are present in the mobile number.")
else:
    print("Invalid input. Please enter a 10-digit mobile number.")
```

```
Nafeesath Neema
Roll n0:42
batch no:MCA-2022-24
Enter a 10-digit mobile number: 9846991683
Absent digits: 5, 0, 2, 7
```