

DATA SCIENCE

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

```
def is_prime(num):
    if num <= 1:
        return False
    for i in range (2,num):
        if num % i == 0:
            return False
    return True
print("SJC22MCA-2025-FEBIN FATHIMA\nS3MCA")
start=int(input("Enter the starting number: "))
end=int(input("Enter the end number: "))
print("non-prime numbers are: ")
for num in range(start,end+1):
    if not is_prime(num):
        print(num,end=" ")
```

```
SJC22MCA-2025-FEBIN FATHIMA
S3MCA
Enter the starting number: 2
Enter the end number: 10
non-prime numbers are:
4 6 8 9 10
Process finished with exit code 0
```

2. Program to print the first N Fibonacci numbers.

```
print("SJC22MCA-2025-FEBIN FATHIMA\nS3MCA")
n=int(input("Enter the limit: "))
a=0
b=1
for i in range(n):
    print(a,end=" ")
    a,b=b,a+b
```

SJC22MCA-2025-FEBIN FATHIMA

S3MCA

Enter the limit: 7

0 1 1 2 3 5 8

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

```
print("SJC22MCA-2025-FEBIN FATHIMA\nS3MCA")
s1=float(input("Enter first side: "))
s2=float(input("Enter second side: "))
s3=float(input("Enter third side: "))
if s1==s2==s3:
    print("Triangle is equilateral")
elif s1==s2 or s1==s3 or s2==s3 :
    print("Triangle is Issociless")
else:
    print("Triangle is Scalene")
```

SJC22MCA-2025-FEBIN FATHIMA

S3MCA

Enter first side: 8

Enter second side: 8

Enter third side: 8

Triangle is equilateral

SJC22MCA-2025-FEBIN FATHIMA

S3MCA

Enter first side: 7

Enter second side: 7

Enter third side: 9

Triangle is Issociless

SJC22MCA-2025-FEBIN FATHIMA

S3MCA

Enter first side: 15

Enter second side: 34

Enter third side: 32

Triangle is Scalene

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

```
def coprime(a,b):
    while b:
        a,b=b,a%b
    return a==1
num1=int(input("Enter first number: "))
num2=int(input("Enter second number: "))
if coprime(num1,num2):
    print("Numbers are coprime")
else:
    print("numbers are not coprime")
```

```
SJC22MCA-2025-FEBIN FATHIMA
S3MCA
```

```
Enter first number: 11
Enter second number: 12
Numbers are coprime
```

```
SJC22MCA-2025-FEBIN FATHIMA
S3MCA
```

```
Enter first number: 10
Enter second number: 15
numbers are not coprime
```

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

```
import cmath
def quadratic_roots(a,b,c):
    disc=b**2 - 4*a*c
    root1=(-b + cmath.sqrt(disc)) / (2*a)
    root2=(-b - cmath.sqrt(disc)) / (2*a)
    root1 = round(root1.real,2) + round(root1.imag,2) * 1j
    root2 = round(root2.real,2) + round(root2.imag,2) * 1j
    return root1,root2
print("SJC22MCA-2025-FEBIN FATHIMA\nS3MCA")
a=float(input("Enter the coefficient of a: "))
```

```

b=float(input("Enter the coefficient of b: "))
c=float(input("Enter the coefficient of c: "))
roots = quadratic_roots(a,b,c)
print("Roots of the quadratic equation are: ",roots)

```

SJC22MCA-2025-FEBIN FATHIMA

S3MCA

Enter the coefficient of a: 2

Enter the coefficient of b: 3

Enter the coefficient of c: 4

Roots of the quadratic equation are: ((-0.75+1.2j), (-0.75-1.2j))

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

```

print("SJC22MCA-2025-FEBIN FATHIMA\nS3MCA")
def perfect_number(num):
    if num <= 0:
        return False

    factor_sum = 0
    for i in range(1, num):
        if num % i == 0:
            factor_sum += i

    print("The sum of factors:", factor_sum)
    return factor_sum == num

num = int(input("Enter a number: "))
if perfect_number(num):
    print(num,"is a perfect number")
else:
    print(num,"is not a perfect number")

```

SJC22MCA-2025-FEBIN FATHIMA

S3MCA

Enter a number: 28

The sum of factors: 28

28 is a perfect number

7. Program to display amstrong numbers upto 1000

```
print("SJC22MCA-2025-FEBIN FATHIMA\nS3MCA")
def armstrong(num):
    num_str = str(num)
    num_digits = len(num_str)
    digit_sum = sum(int(digit) ** num_digits for digit in num_str)
    return digit_sum == num
for num in range (1,1001):
    if armstrong(num):
        print(num)
```

SJC22MCA-2025-FEBIN FATHIMA
S3MCA

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("SJC22MCA-2025-FEBIN FATHIMA\nS3MCA")
```

```
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))
```

SJC22MCA-2025-FEBIN FATHIMA

S3MCA

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): {'Sunday', 'Thursday', 'Tuesday', 'Wednesday', 'Saturday', 'Monday', 'Friday'}

Type of Set: <class 'set'>

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

```

print("SJC22MCA-2025-FEBIN FATHIMA\nS3MCA")
list1=input("Enter list1 elements seperated by space: ").split()
list2=input("Enter list2 elements seperated by space: ").split()
list1 = [int(x) for x in list1]
list2 = [int(x) for x in list2]
result=[]
if len(list1) == len(list2):
    for i in range(len(list1)):
        result.append(list1[i] + list2[i])
    print("After adding list1 and list2 is: ",result)
else:
    print("Addition can be done only when both list have same length")

```

SJC22MCA-2025-FEBIN FATHIMA

S3MCA

Enter list1 elements seperated by space: 1 2 3 4

Enter list2 elements seperated by space: 9 8 6 5

After adding list1 and list2 is: [10, 10, 9, 9]

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

```

print("SJC22MCA-2025-FEBIN FATHIMA\nS3MCA")
rows = int(input("Enter the number of rows: "))
cols = int(input("Enter the number of columns: "))

print("Enter values for the first matrix:")
matrix1 = [[int(input()) for _ in range(cols)] for _ in range(rows)]

print("Enter values for the second matrix:")
matrix2 = [[int(input()) for _ in range(cols)] for _ in range(rows)]

result_matrix = [[matrix1[i][j] + matrix2[i][j] for j in range(cols)] for i in range(rows)]

print("Sum of the matrices:")
for row in result_matrix:
    print(row)

```

```
SJC22MCA-2025-FEBIN FATHIMA
S3MCA
Enter the number of rows: 2
Enter the number of columns: 2
Enter values for the first matrix:
2
3
4
5
Enter values for the second matrix:
3
5
3
2
Sum of the matrices:
[5, 8]
[7, 7]
```

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

```
print("SJC22MCA-2025-FEBIN FATHIMA\nS3MCA")
def bubble_sort(arr):
    n = len(arr)
    for i in range(n):
        for j in range(0, n - i - 1):
            if arr[j] > arr[j + 1]:
                arr[j], arr[j + 1] = arr[j + 1], arr[j]

elements = list(map(int, input("Enter elements separated by spaces:
").split()))

bubble_sort(elements)

print("Sorted elements:", elements)
```



```
SJC22MCA-2025-FEBIN FATHIMA
S3MCA
Enter elements separated by spaces: 2 7 4 9 3 6 1
Sorted elements: [1, 2, 3, 4, 6, 7, 9]

Process finished with exit code 0
|
```

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

```
print("SJC22MCA-2025-FEBIN FATHIMA\nS3MCA")
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_counts = count_vowels(input_string)

print("Vowel counts in the string:")
for vowel, count in vowel_counts.items():
    print(f"{vowel}: {count}")
```

```
SJC22MCA-2025-FEBIN FATHIMA
S3MCA
Enter a string: hello world
Vowel counts in the string:
a: 0
e: 1
i: 0
o: 2
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: $256 - 2 + 5 + 6 = 13$

$256 - 13 = 243$

$243 - 9 = 232 \dots\dots\dots$

```
print("SJC22MCA-2025-FEBIN FATHIMA\nS3MCA")
def sum_of_digits(number):
    return sum(map(int, str(number)))
number = int(input("Enter a positive number: "))

while number > 0:
    digit_sum = sum_of_digits(number)
    print(f"{number} - {digit_sum} = {number - digit_sum}")
    number -= digit_sum
```

SJC22MCA-2025-FEBIN FATHIMA

S3MCA

Enter a positive number: 256

256 - 13 = 243

243 - 9 = 234

234 - 9 = 225

225 - 9 = 216

216 - 9 = 207

207 - 9 = 198

198 - 18 = 180

180 - 9 = 171

171 - 9 = 162

162 - 9 = 153

153 - 9 = 144

144 - 9 = 135

135 - 9 = 126

126 - 9 = 117

117 - 9 = 108

108 - 9 = 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

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("SJC22MCA-2025-FEBIN FATHIMA\nS3MCA")
def find_absent_digits(mobile_number):
    all_digits = set('0123456789')
    mobile_digits = set(mobile_number)
    absent_digits = all_digits - mobile_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("Digits absent in the mobile number:", '
'.join(sorted(absent_digits)))
    else:
        print("All digits are present in the mobile number.")
else:
    print("Invalid input. Please enter a 10-digit mobile number.")
```

SJC22MCA-2025-FEBIN FATHIMA

S3MCA

Enter a 10-digit mobile number: 9947457701

Digits absent in the mobile number: 2 3 6 8