

CYCLE 1

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

Program code

```
print("MUHAMMED HISHAM KP , 41 , MCA-2022-24")
u = int(input("enter lower limit:"))
v = int(input("enter upper limit:"))
print("prime numbers between",u,"and",v,"are:")
for i in range(u, v + 1):
    if i > 1:
        for j in range(2, i):
            if (i % j) == 0:
                print(i)
                break
```

Output

```
MUHAMMED HISHAM KP , 41 , MCA-2022-24
enter lower limit:1
enter upper limit:10
prime numbers between 1 and 10 are:
4
6
8
9
10
```

2. Program to print the first N Fibonacci numbers.

Program code

```
print("MUHAMMED HISHAM KP , 41 , MCA-2022-24")
num = int(input("enter limit:"))
n1, n2 = 0, 1
for i in range(0, num):
    print(n1, end=" ")
    n3 = n1 + n2
    n1 = n2
    n2 = n3
```

Output

```
MUHAMMED HISHAM KP , 41 , MCA-2022-24
enter limit:12
0 1 1 2 3 5 8 13 21 34 55 89
```

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

Program code

```
print("MUHAMMED HISHAM KP , 41 , MCA-2022-24")  
  
print("Input lengths of the triangle sides: ")  
  
a = int(input("a: "))  
b = int(input("b: "))  
c = int(input("c: "))  
  
if a == b == c:  
    print("It is an Equilateral triangle")  
elif a == b or b == c or c == a:  
    print("It is an Isosceles triangle")  
else:  
    print("It is a Scalene triangle")
```

Output

```
MUHAMMED HISHAM KP , 41 , MCA-2022-24  
Input lengths of the triangle sides:  
a: 3  
b: 3  
c: 3  
It is an Equilateral triangle
```

```
MUHAMMED HISHAM KP , 41 , MCA-2022-24  
Input lengths of the triangle sides:  
a: 4  
b: 5  
c: 4  
It is an Isosceles triangle
```

```
MUHAMMED HISHAM KP , 41 , MCA-2022-24  
Input lengths of the triangle sides:  
a: 7  
b: 5  
c: 4  
It is a Scalene triangle
```

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

Program code

```
print("MUHAMMED HISHAM KP , 41 , 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:"))  
second = int(input("enter first number:"))  
  
if are_coprime(first, second):  
    print("%d and %d are coprime" % (first, second))  
else:  
    print("%d and %d are not coprime" % (first, second))
```

Output

```
MUHAMMED HISHAM KP , 41 , MCA-2022-24  
enter first number:11  
enter first number:17  
11 and 17 are coprime
```

```
MUHAMMED HISHAM KP , 41 , MCA-2022-24  
enter first number:12  
enter first number:18  
12 and 18 are not coprime
```

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

Program code

```
print("MUHAMMED HISHAM KP , 41 , 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

dl = d**0.5

if d < 0:

    print("the roots are imaginary")

else:

    r1 = (-b + dl) / 2 * a

    r2 = (-b - dl) / 2 * a

    print("the first root:", round(r1, 2))

    print("the second root:", round(r2, 2))
```

Output

```
MUHAMMED HISHAM KP , 41 , MCA-2022-24
Equation: ax^2 + bx + c
enter a:5
enter b:7
enter c:3
the roots are imaginary
```

```
MUHAMMED HISHAM KP , 41 , MCA-2022-24
Equation: ax^2 + bx + c
enter a:1
enter b:-3
enter c:2
the first root: 2.0
the second root: 1.0
```

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

Program code

```
print("MUHAMMED HISHAM KP , 41 , MCA-2022-24")  
num = int(input("enter a number:"))  
sum = 0  
for i in range(1, num):  
    if (num % i == 0):  
        sum = sum + i  
if (sum == num):  
    print("the entered number is perfect number")  
else:  
    print("the entered number is not a perfect number")
```

Output

```
MUHAMMED HISHAM KP , 41 , MCA-2022-24  
enter a number:6  
the entered number is perfect number
```

```
MUHAMMED HISHAM KP , 41 , MCA-2022-24  
enter a number:45  
the entered number is not a perfect number
```

7. Program to display armstrong numbers upto 1000

Program code

```
print("MUHAMMED HISHAM KP , 41 , MCA-2022-24")
print("armstrong numbers upto 1000 are:")
for num in range(1, 1001):
    num_digit = len(str(num))
    sum = 0
    temp = num
    while temp > 0:
        digit = temp % 10
        sum +=digit ** num_digit
        temp //=10
    if num == sum:
        print(num)
```

Output

```
MUHAMMED HISHAM KP , 41 , MCA-2022-24
armstrong numbers upto 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.

Program code

```
print("MUHAMMED HISHAM KP , 41 , MCA-2022-24")

days_list = ["monday", "tuesday",
"wednesday", "Thursday", "Friday", "saturday", "sunday"]

print("list:", days_list)

print("Type of list:", type(days_list))

days_tuple = ("monday", "tuesday",
"wednesday", "Thursday", "Friday", "saturday", "sunday")

print("Tuple:", days_list)

print("Type of list:", type(days_tuple))

days_dict = {1: "monday", 2: "tuesday", 3: "wednesday", 4: "Thursday", 5: "Friday",
6: "saturday", 7: "sunday"}

print("Tuple:", days_dict)

print("Type of list:", type(days_dict))

days_set =
{"monday", "tuesday", "wednesday", "Thursday", "Friday", "saturday", "sunday"}

print("Tuple:", days_set)

print("Type of list:", type(days_set))
```

Output

```
MUHAMMED HISHAM KP , 41 , MCA-2022-24
list: ['monday', 'tuesday', 'wednesday', 'Thursday', 'Friday', 'saturday', 'sunday']
Type of list: <class 'list'>
Tuple: ['monday', 'tuesday', 'wednesday', 'Thursday', 'Friday', 'saturday', 'sunday']
Type of list: <class 'tuple'>
Tuple: {1: 'monday', 2: 'tuesday', 3: 'wednesday', 4: 'Thursday', 5: 'Friday', 6: 'saturday', 7: 'sunday'}
Type of list: <class 'dict'>
Tuple: {'monday', 'sunday', 'Thursday', 'wednesday', 'tuesday', 'saturday', 'Friday'}
Type of list: <class 'set'>
```


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

Program code

```
print("MUHAMMED HISHAM KP , 41 , MCA-2022-24")
input_str1 = input("Enter elements list1 separated by spaces:")
list1 = input_str1.split()

input_str2 = input("Enter elements list2 separated by spaces:")
list2 = input_str2.split()
if len(list1) != len(list2):
    print("Error:List 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)
```

Output

```
MUHAMMED HISHAM KP , 41 , MCA-2022-24
Enter elements list1 separated by spaces:1 2 3 4
Enter elements list2 separated by spaces:5 6 7 8
Element-wise sum of the two lists: [6, 8, 10, 12]
```

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

Program code

```
print("MUHAMMED HISHAM KP , 41 , MCA-2022-24")

def input_matrix(rows, cols):
    matrix = []
    for i in range(rows):
        row = []
        for j in range(cols):
            element = int(input("Enter the element: "))
            row.append(element)
        matrix.append(row)
    return matrix

def add_matrices(matrix1, matrix2):
    if len(matrix1) != len(matrix2) or len(matrix1[0]) != len(matrix2[0]):
        return None
    result = []
    for i in range(len(matrix1)):
        row = []
        for j in range(len(matrix1[0])):
            element = matrix1[i][j] + matrix2[i][j]
            row.append(element)
        result.append(row)
    return result

def display_matrix(matrix):
    for row in matrix:
        for element in row:
            print(element, end=" ")
        print()

rows = int(input("Enter the number of rows: "))
```

```
cols = int(input("Enter the number of columns: "))

print("Enter elements of the first matrix:")

matrix1 = input_matrix(rows, cols)

print("Enter elements of the second matrix:")

matrix2 = input_matrix(rows, cols)

result_matrix = add_matrices(matrix1, matrix2)

if result_matrix:

    print("\nMatrix 1:")

    display_matrix(matrix1)

    print("\nMatrix 2:")

    display_matrix(matrix2)

    print("\nSum of the matrices:")

    display_matrix(result_matrix)

else:

    print("Matrix addition is not possible. Matrices are of different sizes.")
```

Output

```
MUHAMMED HISHAM KP , 41 , MCA-2022-24
Enter the number of rows: 2
Enter the number of columns: 2
Enter elements of the first matrix:
Enter the element: 1
Enter the element: 3
Enter the element: 5
Enter the element: 6
Enter elements of the second matrix:
Enter the element: 3
Enter the element: 5
Enter the element: 7
Enter the element: 8

Matrix 1:
1 3
5 6

Matrix 2:
3 5
7 8

Sum of the matrices:
4 8
12 14
```

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

Program code

```
print("MUHAMMED HISHAM KP , 41 , 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)
```

Output

```
MUHAMMED HISHAM KP , 41 , MCA-2022-24
Enter elements : 7 4 3 8 2 4 1
Original List: [7, 4, 3, 8, 2, 4, 1]
Sorted List: [1, 2, 3, 4, 4, 7, 8]
```

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

Program code

```
print("MUHAMMED HISHAM KP , 41 , 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}")
```

Output

```
MUHAMMED HISHAM KP , 41 , MCA-2022-24
Enter a string: welcome to my home
a: 0
e: 3
i: 0
o: 3
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: $256 \rightarrow 2+5+6=13$

$$256-13=243$$

$$243-9=232\ldots\ldots$$

Program code

```
print("MUHAMMED HISHAM KP , 41 , 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.")
```

Output

```
MUHAMMED HISHAM KP , 41 , MCA-2022-24
Enter a positive number: 25
25 - 7 = 18
The number is now positive or zero.
18 - 9 = 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

Program code

```
print("MUHAMMED HISHAM KP , 41 , 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.")
```

Output

```
MUHAMMED HISHAM KP , 41 , MCA-2022-24
Enter a 10-digit mobile number: 7034642606
Absent digits: 9, 8, 1, 5
```