

Raqib Ali Roll No:333134

1. Write a Python function to find the maximum of three numbers.

```
def maximum(a,b,c):  
    return max(a,b,c)  
  
num1=int(input("Enter a number: "))  
num2=int(input("Enter a number: "))  
num3=int(input("Enter a number: "))  
number=maximum(num1,num2,num3)  
  
print(f"The maximum in {num1}, {num2} and {num3} is {number}.")  
  
Enter a number: 10  
Enter a number: 15  
Enter a number: 20  
  
The maximum in 10, 15 and 20 is 20.
```

2. Write a Python function to sum all the numbers in a list.
Sample List : (8, 2, 3, 0, 7)

```
def sumnumber(numbers):  
    return sum(numbers)  
  
sample_list=[8,2,3,0,7]  
number=sumnumber(sample_list)  
  
print(f"The sum of following list is {number}")  
  
The sum of following list is 20
```

3. Write a Python function to multiply all the numbers in a list. Sample List : (8, 2, 3, -1, 7)

```
def multiplenumbers(numbers):  
    result=1  
    for num in numbers:  
        result*=num  
    return result  
  
sample_list=[8,2,3,-1,7]
```

```
number=multiplenumbers(sample_list)

print(f"The multiple of {sample_list} is {number}.")

The multiple of [8, 2, 3, -1, 7] is -336.
```

4. Write a Python program to reverse a string. Sample String : "1234abcd"

```
def reverse_str(string):
    return string[::-1]

sample_string="1234abcd"

stringreverse=reverse_str(sample_string)

print(f"The reverse of {sample_string} is {stringreverse}.")

The reverse of 1234abcd is dcba4321.
```

5. Write a Python function to calculate the factorial of a number (a non-negative integer). The function accepts the number as an argument

```
def factorial(n):
    # Check if the input is a non-negative integer
    if n < 0:
        return "Factorial is not defined for negative numbers"
    elif n == 0 or n == 1:
        return 1
    else:
        result = 1
        for i in range(2, n + 1):
            result *= i
        return result

number=int(input("Enter a number: "))
num=factorial(number)

print(f"The facotrial of {num} is {number}.")

Enter a number: 6

The facotrial of 720 is 6.
```

6. Write a Python function to check whether a number falls within a given range.

```
def given_range(num, start_range, end_range):  
    if num in range(start_range, end_range):  
        print(f"The number is within {start_range} and {end_range}.")  
    else:  
        print(f"The given number {num} is not in range.")  
  
number=int(input("Enter a number: "))  
ras=int(input("Enter a start range: "))  
rae=int(input("Enter an end range: "))  
  
check=given_range(number, ras, rae)  
  
Enter a number: 12  
Enter a start range: 10  
Enter an end range: 15  
  
The number is within 10 and 15.
```

7. Write a Python function that accepts a string and counts the number of upper and lower case letters. Sample String : 'The quick Brow Fox'

Expected Output :

No. of Upper case characters : 3

No. of Lower case Characters : 12

```
def upper_lower(string):  
    upper_case=0  
    lower_case=0  
    for char in string:  
        if char.isupper():  
            upper_case+=1  
        elif char.islower():  
            lower_case+=1  
  
    return upper_case, lower_case  
  
string=input("Enter a sentence: ")  
  
check=upper_lower(string)
```

```
print(check)
```

```
Enter a sentence: A quick Brown Fox jumps over a dog.
```

```
(4, 23)
```

8. Write a Python function that takes a list and returns a new list with distinct elements from the first list.

Sample List : [1,2,3,3,3,3,4,5]

Unique List : [1, 2, 3, 4, 5]

```
def get_unique_elements(input_list):  
    return list(set(input_list))  
  
liste=[]  
  
while True:  
    try:  
        inp=int(input("Enter a number or press q to Quit: "))  
        if inp=="q":  
            break  
        else:  
            liste.append(inp)  
    except ValueError:  
        print("Quitting...")  
        break  
  
single=get_unique_elements(liste)  
  
print(f"The given list was {liste}. The list with distinct elements is {single}.")
```

```
Enter a number or press q to Quit: 1  
Enter a number or press q to Quit: 1  
Enter a number or press q to Quit: 2  
Enter a number or press q to Quit: 5  
Enter a number or press q to Quit: 4  
Enter a number or press q to Quit: 7  
Enter a number or press q to Quit: 8  
Enter a number or press q to Quit: 9  
Enter a number or press q to Quit: q
```

```
Quitting...
```

```
The given list was [1, 1, 2, 5, 4, 7, 8, 9]. The list with distinct  
elements is [1, 2, 4, 5, 7, 8, 9].
```

9. Write a Python function that takes a number as a parameter and checks whether the number is prime or not.

Note : A prime number (or a prime) is a natural number greater than 1 and that has no positive divisors other than 1 and itself.

```
def is_prime(number):  
    # Check if the number is less than or equal to 1  
    if number <= 1:  
        return False  
  
    # Check for divisibility from 2 to the square root of the number  
    for i in range(2, int(number**0.5) + 1):  
        if number % i == 0:  
            return False # If divisible by any number, it's not prime  
  
    return True # If no divisors were found, the number is prime  
  
inp=int(input("Enter a number: "))  
check=is_prime(inp)  
print(f"Is {inp} a prime number??? {check}.")  
Enter a number: 2  
Is 2 a prime number??? True.
```

10. Write a Python program to print the even numbers from a given list.

Sample List : [1, 2, 3, 4, 5, 6, 7, 8, 9]

Expected Result : [2, 4, 6, 8]

```
def even_num(lst3):  
    lst=[]  
    for i in range(len(lst3)):  
        if i%2==0:  
            lst.append(i)  
    return (lst)  
  
liste=[]  
while True:
```

```

try:
    inp=int(input("Enter a number or press q to Quit: "))
    if inp=="q":
        break
    else:
        liste.append(inp)
except ValueError:
    print("Quitting...")
    break

check=even_num(liste)

print(f"The even numbers in the list {liste} is {check}.")

Enter a number or press q to Quit: 1
Enter a number or press q to Quit: 2
Enter a number or press q to Quit: 4
Enter a number or press q to Quit: 5
Enter a number or press q to Quit: 8
Enter a number or press q to Quit: 7
Enter a number or press q to Quit: 9
Enter a number or press q to Quit: 6
Enter a number or press q to Quit: q

Quitting...
The even numbers in the list [1, 2, 4, 5, 8, 7, 9, 6] is [0, 2, 4, 6].

```

11. Write a Python function to check whether a number is "Perfect" or not

```

def is_perfect(number):
    if number <= 0:
        return False # Perfect numbers are positive integers

    divisors_sum = 0
    # Find divisors of the number (excluding the number itself)
    for i in range(1, number):
        if number % i == 0:
            divisors_sum += i

    # Check if the sum of divisors equals the number
    return divisors_sum == number

# Example usage:
num = int(input("Enter a number: "))
if is_perfect(num):
    print(f"{num} is a perfect number.")
else:
    print(f"{num} is not a perfect number.")

```

Enter a number: 5

5 is not a perfect number.

12. Write a Python function that checks whether a passed string is a palindrome or not.

```
import re

def is_palindrome(s):
    # Remove all non-alphanumeric characters and convert to lowercase
    cleaned_string = re.sub(r'^a-zA-Z0-9', '', s).lower()

    # Check if the string is equal to its reverse
    return cleaned_string == cleaned_string[::-1]

# Example usage:
string = input("Enter a word: ")
if is_palindrome(string):
    print(f"'{string}' is a palindrome.")
else:
    print(f"'{string}' is not a palindrome.")
```

Enter a word: Dog

'Dog' is not a palindrome.