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

In [1]:
```
def max_num(a,b,c):
    return a,b,c
```

In [4]:
```
max_num(2,4,5)
```

Out[4]:
```
(2, 4, 5)
```

2. Write a Python function to sum all the numbers in a list. Sample List : (8, 2, 3, 0, 7) Expected Output : 20

In [26]:
```
def sum_list(lst):
    s=0
    for num in lst:
        s+=num
    return s
```

In [28]:
```
sum_list([8,2,3,0,7])
```

Out[28]:
```
20
```

3. Write a Python function to multiply all the numbers in a list. Sample List : (8, 2, 3, -1, 7) Expected Output : -336

In [58]:
```
def multiply(lst):
    s=1
    for num in lst:
        s*=num
    print(s)
```

In [59]:
```
multiply([8, 2, 3, -1, 7])
```

```
-336
```

4. Write a Python program to reverse a string. Sample String : "1234abcd" Expected Output : "dcba4321"

In [94]:
```
def reverse(string):
    strings="1234abcd"
    print(strings[::-1])
```

In [95]:
```
reverse("")
```

```
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.

In [10]:
```
def facturial():
    n=10
    for i in range(1,n):
        n*=i
    return n
facturial()
```

Out[10]:
```
3628800
```

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

In [12]:
```
def is_within_range(num,start, end):
     return start <= num <= end

# Example usage:
number = 20
start = 2
end = 290
print(is_within_range(number, start, end))
```

```
True
```

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

In [1]:
```
def count_case_letter(input_string):
    upper_case_count = sum(1 for char in input_string if char.isupper())
    lower_case_count = sum(1 for char in input_string if char.islower())
    return upper_case_count, lower_case_count

string="The quick Brow Fox"
upper_count,lower_count=count_case_letter(string)
print(f"the upper case letter  is :{upper_count}")
print(f"the upper case letter  is :{lower_count}")
```

```
the upper case letter  is :3
the upper case letter  is :12
```

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]

In [52]:
```
def unique_list(input_list):
    return list(set(input_list))
simple_list=[1,2,3,3,3,4,5]
unique_list=unique_list(simple_list)
print("Sample List:", simple_list)
print("Unique List:", unique_list)
```

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

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.

In [62]:
```
def prime_num(input_num):
    if input_num<=1:
        return "enter greter then 1"
    for i in range(2,input_num):
        if input_num%i==0:
            return f"it is not prime no:{input_num}"
    return f"it is a prime no:{input_num}"
prime_num(7)
```

Out[62]:
```
'it is a prime no:7'
```

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]

In [49]:
```
def even():
    list1=[ 2, 3, 4, 5, 6, 7, 8, 9]
    list2=list1[::2]
    return list2
even()
```

Out[49]:
```
[2, 4, 6, 8]
```

11. Write a Python function to check whether a number is "Perfect" or not. According to Wikipedia : In number theory, a perfect number is a positive integer that is equal to the sum of its proper positive divisors, that is, the sum of its positive divisors

excluding the number itself (also known as its aliquot sum). Equivalently, a perfect number is a number that is half the sum of all of its positive divisors (including itself). Example : The first perfect number is 6, because 1, 2, and 3 are its proper positive divisors, and 1 + 2 + 3 = 6. Equivalently, the number 6 is equal to half the sum of all its positive divisors: (1 + 2 + 3 + 6) / 2 = 6. The next perfect number is 28 = 1 + 2 + 4 + 7 + 14. This is followed by the perfect numbers 496 and 8128.

In [1]:
```python
def is_perfect_number(n):

    if n <= 1:
        return False  # Perfect numbers are positive integers greater than 1

    # Find the sum of proper divisors (excluding the number itself)
    divisors_sum = sum(i for i in range(1, n // 2 + 1) if n % i == 0)

    return divisors_sum == n


# Test the function
test_numbers = [6, 28, 496, 8128, 10, 12]
for num in test_numbers:
    print(f"{num} is a perfect number: {is_perfect_number(num)}")
```

```
6 is a perfect number: True
28 is a perfect number: True
496 is a perfect number: True
8128 is a perfect number: True
10 is a perfect number: False
12 is a perfect number: False
```

12. Write a Python function that checks whether a passed string is a palindrome or not. Note: A palindrome is a word, phrase, or sequence that reads the same backward as forward, e.g., madam or nurses run.

In [21]:
```
def palindrome(string):
    strings="madam"
    if strings=="madam":
        print("yes it is palindrome",strings )
    else:
        print("it is not palindrome",strings)
    print(strings[::-1])

palindrome("")
```

```
yes it is palindrome madam
madam
```

In [ ]: