1. Function to find Maximum of three numbers.

```
def max_of_three(a,b,c):
  if a>b and a>c:
    print(f'{a} is the maximum number')
  elif b>a and b>c:
    print(f'{b} is the maximum number')
  else :
    print(f'{c} is the maximum number')
max_of_three(int(input()),int(input()),int(input()))
```

```
5
10
3
10 is the maximum number
```

2.Funtion to Sum Numbers in a list

+ Code + Text

```
def sum_list(lst):
  total = 0
  for i in lst:
    total=total+i
  return total
print(sum_list([8,2,3,0,7]))
```

```
20
```

```
def sum_list(lst):
  total = 0
  for i in lst:
    total=total+i
  print(total)
sum_list([8,2,3,0,7])
```

```
20
None
```

Using recursion

```
def sum_list(lst):
    if len(lst) == 1:  # Base case: If the list has only one element
        return lst[0]  # Return that element
    else:
        return lst[0] + sum_list(lst[1:])  # Sum the first element with the sum of the rest of the list

print(sum_list([8, 2, 3, 0, 7]))  # Output: 20
```

```
20
```

Errors

```
def sum_list(lst):
  total = 0
  for i in lst:
    total=total+i
    return total
print(sum_list([8,2,3,0,7]))
```

```
8
```

```
def sum_list(lst):
  total = 0
  for i in lst:
    total=total+i
    print(total)
sum_list([8,2,3,0,7])
```

```
8
10
13
13
20
```

```
def sum_list(lst):
  total = 0
  for i in lst:
    total=total+i
  print(total)
print(sum_list([8,2,3,0,7]))
```

```
20
None
```

3.Funtion to Multiply Numbers in a list

```
def sum_list(lst):
  total = 1
  for i in lst:
    total=total*i
  return total
print(sum_list([8,2,3,-1,7]))
```

```
-336
```

```
def sum_list(lst):
  total = 1
  for i in lst:
    total=total*i
  print(total)
sum_list([8,2,3,-1,7])
```

```
-336
```

```
def sum_list(lst):
    if len(lst) == 1:
        return lst[0]
    else:
        return lst[0] * sum_list(lst[1:])

print(sum_list([8,2,3,-1,7]))
```

```
-336
```

4.Function to Reverse a string

```
def rev(a):
  str = ''
  for i in a:
    str = i + str
  return str
print(rev(input()))
```

```
1234abcd
dcba4321
```

```
def reverse_string(s):
    # Base case: if the string is empty or has only one character, return the string
    if len(s) <= 1:
        return s
    # Recursive case: reverse the substring starting from the second character and concatenate the first character
    return reverse_string(s[1:]) + s[0]

# Test the function
input_str = input("Enter a string: ")
print("Reversed string:", reverse_string(input_str))
```

5.Function to Calculate Factorial

```
def factorial(a):
  total = 1
  for i in range(1,a+1):
    total=total*i
  return total
print(factorial(int(input())))
```

```
5
120
```

```
def factorial(a):
  if a<=1:
    return 1
  else:
    return a*factorial(a-1)
print(factorial(int(input())))
```

```
5
120
```

6.Funtion to Check Number within Range

```
def in_range(a,b,c):
  if a in range(b,c):
    return True
  else:
    return False
print(in_range(int(input()),int(input()),int(input())))
```

```
5
1
10
True
```

```
n=int(input())
b=int(input())
a=int(input())
def range():
  if a<=n<b:
    print('true')
  else:
    print('false')
range()
```

Problem

7.Function to count Upper and Lower Case Characters

```
def count_case(a):
  lower_count = 0
  upper_count = 0
  for i in a:
    if i.islower():
      lower_count+=1
    elif i.isupper():
      upper_count +=1
  print(f'No. of Upper case character: {upper_count},No. of Lower case character: {lower_count}')
count_case('The quick Brown Fox')
```

```
 No. of Upper case character: 3,No. of Lower case character: 13
```

```
def count_case():
  y=input()
  lower_count=0
  upper_count=0
  for x in y:
    count=0
    if x.lower() in y:
        lower_count+=1
    elif x.upper() in y:
        upper_count+=1
  print('no. of upper case character is:',upper_count,'no. of lower case character is:',lower_count)
count_case()
```

```
def count_upper_lower(string):
    # Initialize counters for uppercase and lowercase characters
    upper_count = 0
    lower_count = 0

    # Iterate through each character in the string
    for char in string:
        # Check if the character is uppercase
        if char.isupper():
            upper_count += 1
        # Check if the character is lowercase
        elif char.islower():
            lower_count += 1

    # Return the counts of uppercase and lowercase characters
    return upper_count, lower_count

# Test the function
input_string = input("Enter a string: ")
upper, lower = count_upper_lower(input_string)
print("Uppercase characters:", upper)
print("Lowercase characters:", lower)
```

8.Function to Return Unique Elements from a List

```
def unique_elements(lst):
  a = set(lst)
  b = list(a)
  return b
print(unique_elements([1,2,3,3,3,4,5]))
```

```
 [1, 2, 3, 4, 5]
```

```
def unique_elements(lst):
  A = []
  for i in lst:
    if i in A:
      continue
    else:
      A.append(i)
  return A
print(unique_elements([1,2,3,3,3,4,5]))
```

```
 [1, 2, 3, 4, 5]
```

9.Function to check Prime Number

```
def is_prime(a):
  if a==0 or a==1:
    return True
  elif a==2:
    return False
  else:
    for i in range(2,a):
      if a%i == 0:
        return False
      else:
        return True
print(is_prime(int(input())))
```

```
7
True
```

10. Program to print Even Numbers from a list

```
def even_numbers(lst):
  A=[]
  for i in lst:
    if i%2 == 0:
      A.append(i)
    else:
      continue
  return A
print(even_numbers([1,2,3,4,5,6,7,8,9]))
```

```
[2, 4, 6, 8]
```

11.Function to check Perfect Number

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.

```
n=int(input())
def is_perfect(a):
  sum=0
  if a>0:
    for i in range(1,a):
      if a%i==0:
        sum = sum+i
    if sum == a:
      return True
    else:
      return False
  else:
    return False
print(is_perfect(n))
```

12.Function to check Palindrome

```
def is_palindrome(a):
  str=''
  for i in a:
    str=i+str
  if str==a:
    return True
  else:
    return False
print(is_palindrome(input()))
```

```
 sir
 False
```