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

```
def find_max(num1,num2,num3):
    if num1>num2 and num1>num3:
        return num1
    elif num2>num1 and num2>num3:
        return num2
    else:
        return num3

find_max(12,56,78)
```

```
78
```

2. Write a Python function to sum all the numbers in a list.

```
Sample List : (8, 2, 3, 0, 7)
Expected Output : 20
```

```
def sum_num(num1,num2,*num):
    nem=0
    for a in num:
        nem+=a
        x = nem+num1+num2
    return x

sum_num(8,2,3,0,7)
```

```
20
```

3. Write a Python function to multiply all the numbers in a list.

```
Sample List : (8, 2, 3, -1, 7)
Expected Output : -336
```

```
def multi_num(lst):
    x=1
    for a in lst:
        x*=a
    return x

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

```
-336
```

4. Write a Python program to reverse a string.

```
Sample String : "1234abcd"
Expected Output : "dcba4321"
```

```
def rev_str(string):
        string=string[::-1]
        return string

rev_str("1234abcd")
```

```
'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 fact(fac):
    tot=1
    if fac>0:
        for i in range(1,fac+1):
            tot=tot*i
        return f"The Factorial of {fac} is {tot}"
    elif fac==0:
        return "The factorial of 0 is 1"
    elif fac<0:
        return "Factorial does not exist for negative numbers"


fact(5)
```

```
'The Factorial of 5 is 120'
```

```
fact(-1)
```

```
'Factorial does not exist for negative numbers'
```

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

```
def is_within_range(num,fro,to):
    if num<=to and num>=fro:
        return True
    else:
        return False

is_within_range(7,3,90)
```

```
True
```

```
is_within_range(8,9,99)
```

```
False
```

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 upperlower_count(string):
    num_low=0
    num_up=0
    for a in string:
        if a.islower():
            num_low=num_low+1
        elif a.isupper():
            num_up=num_up+1
    return f"""No. of lower case characters : {num_low}
No. of Upper case characters :{num_up}"""


print(upperlower_count("I am Hammad Baloch thank you"))
```

```
No. of lower case characters : 20
No. of Upper case characters :3
```

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 filter_list(lst):
    b=[]
    for a in lst:
        if a not in b:
            b.append(a)
    return b

filter_list([1,2,3,3,3,3,4,5])
```

```
[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.
```

```
def is_prime(number):
    if number<=1:
        return False
    if number<=3:
        return True
    if number%2==0 or number%3==0:
        return False
    for i in range(5, int(math.sqrt(number))+1,6):
        if number%i==0 or number%(i+2)==0:
            return False
    return True

is_prime(7)
```

```
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(ev):
    even_list=[]
    for a in ev:
        if a<=0:
            return "You must enter real positive numbers"
        for e in ev:
            if e%2==0:
                even_list.append(e)
        return even_list

even([1,2,3,4,5,6,7,8,9])
```

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

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

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.

```
def is_perfect(num):
    tot=0
    for a in range(1,num//2+1):
        if num%a==0:
            tot+=a
```

```
    if tot==num:
        return True
    else:
        return False

is_perfect(6)
```

```
True
```

```
is_perfect(28)
```

```
True
```

```
is_perfect(496)
```

```
True
```

```
is_perfect(10)
```

```
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.

```
def is_palin(pal):
    pal=pal.replace(' ','').lower()
    new_pal=pal[::-1]
    if pal==new_pal:
        return True
    else:
        return False

is_palin("Nurses run")
```

```
True
```

```
is_palin('This cant be a palindrome')
```

```
False
```