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

```
def max3(n1, n2, n3):
  if n1==n2 and n2==n3:
    print(n1, n2, n3, "all three are equal")
  elif n1>n2 and n1>n3:
    print(n1, "is greatest")
  elif n2>n1  and n2>n3:
    print(n2, "is greatest")
  elif n3>n1 and n3>n2:
    print(n3, "is greatest")
  else:
    print("any two are equal")

no1=float(input("Enter no1: "))
no2=float(input("Enter no2: "))
no3=float(input("Enter no3: "))
max3(no1, no2, no3)
```

```
Enter no1: 2
Enter no2: 6
Enter no3: 7
7.0 is greatest
```

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

```
'''
Sample List : (8, 2, 3, 0, 7)
Expected Output : 20
'''
def listsum(larg):
  sum=0
  for ele in larg:
    sum=sum+ele
  return sum

sample=[8, 2, 3, 0, 7]
print("sum: ", listsum(sample))
```

```
sum:  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 listmul(larg):
  product=1
  for ele in larg:
    product=product*ele
  return product

mulsample=[8, 2, 3, -1, 7]
print("Product: ", listmul(mulsample))
```

```
Product:  -336
```

4. Write a Python program to reverse a string.

```
'''
Sample String : "1234abcd"
Expected Output : "dcba4321"
'''
def strreverse(strarg):
  return strarg[::-1]

inputstr="1234abcd"
print(strreverse(inputstr))
```

```
dcba4321
```

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

```
def factorial(num):
  fact=1
  for i in range(num, 0, -1):
    fact=fact*i
  return fact

inputno=int(input("Enter input number: "))
print("Factorial of ", inputno, "is", factorial(inputno))
```

```
Enter input number: 5
Factorial of  5 is 120
```

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

```
def no_in_range(start, end, chk):
  return chk in range(start, end+1)

starting=int(input("Enter start of range: "))
ending=int(input("Enter end of range: "))
chkno=int(input("Enter number to check in range: "))
print(no_in_range(starting, ending, chkno))
```

```
Enter start of range: 1
Enter end of range: 99
Enter number to check in range: 99
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
'''
def count_upper_lower(strarg):
  uppercnt, lowercnt=0, 0
  for ch in strarg:
    if ch.isupper():
      uppercnt+=1
    if ch.islower():
      lowercnt+=1
  return uppercnt, lowercnt

samplestr='The quick Brow Fox'
ucnt, lcnt=count_upper_lower(samplestr)
print("No. of Upper case characters :", ucnt)
print("No. of Lower case characters :", lcnt)
```

```
No. of Upper case characters : 3
No. of Lower case characters : 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]
'''
def unique_ele_list(listarg):
  reslst=[]
  for item in listarg:
    if item not in reslst:
      reslst.append(item)
  return reslst

samplelst=[1,2,3,3,3,3,4,5]
print("Sample List :", samplelst)
print("Unique List :", unique_ele_list(samplelst))
```

```
Sample List : [1, 2, 3, 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.

```
# prime number check
def isprime(num):
  flag=True
  if num==1:
    return flag
  else:
    for i in range(2, num//2+1):
        if num%i==0:
            flag=False
            return flag
    return flag

no=int(input("Enter number to check prime:"))
print("Prime check for", no, ":", isprime(no))
```

```
Enter number to check prime:5
Prime check for 5 : 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 getevens(listarg):
  reslst=[]
  for ele in listarg:
    if ele%2==0:
      reslst.append(ele)
  return reslst

samplelst=[1, 2, 3, 4, 5, 6, 7, 8, 9]
print(getevens(samplelst))
```

```
[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.
'''
def isperfectno(num):
  sum=0
  for i in range(1, (num//2)+1):
    if num%i==0:
      sum=sum+i
  return sum==num

num=int(input("Enter number to check perfect number: "))
print("Perfect number check for ", num, "is", isperfectno(num))
```

```
Enter number to check perfect number: 6
Perfect number check for  6 is True
```

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.

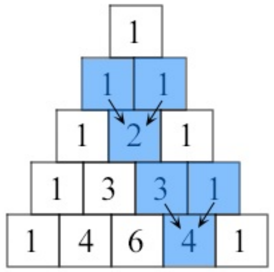
```
def ispallindrome(strarg):
  return strarg==strarg[::-1]

inputstr=input("Enter a string to check pallindrome: ")
print("Pallindrome check for", inputstr, " :", ispallindrome(inputstr))
```

```
Enter a string to check pallindrome: madam
Pallindrome check for madam  : True
```

13. Write a Python function that prints out the first n rows of Pascal's triangle.

Note : Pascal's triangle is an arithmetic and geometric figure first imagined by Blaise Pascal.

Sample Pascal's triangle :



Each number is the two numbers above it added together

```
def pascalt(n):
  for i in range(0, n+1):
    temp=row=11**i
    print(" "*(n-i), end="")
    while temp>0:
      d=temp%10
      temp=temp//10
      print(d, " ", end="")
    print()

pascalt(4)
```

```
     1
    1  1
   1  2  1
  1  3  3  1
 1  4  6  4  1
```

14. Write a Python function to check whether a string is a pangram or not. Note : Pangrams are words or sentences containing every letter of the alphabet at least once.

```
'''
For example : "The quick brown fox jumps over the lazy dog"
'''

def ispangram(strarg):
  strlow=strarg.lower()
  flag=True
  for i in range(ord('a'), ord('z')+1):
    if chr(i) not in strlow:
      flag=False
      break
  return flag

samplestr="The quick brown fox jumps over the lazy dog"
print("Sample string:", samplestr)
print("Pangram check for Sample string :", ispangram(samplestr))
```

```
Sample string: The quick brown fox jumps over the lazy dog
Pangram check for Sample string : True
```

15. Write a Python program that accepts a hyphen-separated sequence of words as input and prints the words in a hyphen-separated sequence after sorting them alphabetically.

```
'''
Sample Items : green-red-yellow-black-white
Expected Result : black-green-red-white-yellow
'''
def sorthyphensep(strarg):
  wlst=strarg.split("-")
  wlst.sort()
  resstr="-".join(wlst)
  return resstr

inpstr="green-red-yellow-black-white"
print(sorthyphensep(inpstr))
```

```
black-green-red-white-yellow
```

16. Write a Python function to create and print a list where the values are the squares of numbers between 1 and 30 (both included).

```python
def sqlstgen():
  res=[]
  for i in range(1, 31):
    res.append(i*i)
  return res

print(sqlstgen())
```

```
[1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169, 196, 225, 256, 289, 324, 361, 400, 441, 484, 529, 576, 625, 676, 729, 784, 841, 900]
```

17. Write a Python program to create a chain of function decorators (bold, italic, underline etc.).

```python
def bold(strarg):
  bold = '\033[1m'
  end = '\033[0m'
  return bold+strarg+end

def italic(strarg):
  italics = '\033[3m'
  end = '\033[0m'
  return italics+strarg+end

def underline(strarg):
  underline = '\033[4m'
  end = '\033[0m'
  return underline+strarg+end

def formatter(strarg, fmting):
  if fmting=="bold":
    strarg=bold(strarg)
  if fmting=="italics":
    strarg=italic(strarg)
  if fmting=="underline":
    strarg=underline(strarg)
  return strarg

inputstr="this is testing string for formatting"
print("Bold:", formatter(inputstr, "bold"))
print("Italic:", formatter(inputstr, "italics"))
print("Underlined:", formatter(inputstr, "underline"))
```

```
Bold: this is testing string for formatting
Italic: this is testing string for formatting
Underlined: this is testing string for formatting
```

18. Write a Python program to execute a string containing Python code.

```python
def pyrunner(code):
  exec(code)

snippet='''a, b=1, 2
print("a:", a, "b:", b)
print("sum of a & b:", a+b)'''
print(snippet, "\n\nAbove Code is executed below: \n")
pyrunner(snippet)
```

```
a, b=1, 2
print("a:", a, "b:", b)
print("sum of a & b:", a+b)

Above Code is executed below:

a: 1 b: 2
sum of a & b: 3
```

19. Write a Python program to access a function inside a function.

```
def menu(a):
  print("From menu")
  def add(b):
    print("From sum")
    print("Sum: ", a+b)
  return add

no1=int(input("Enter a number: "))
no2=int(input("Enter another number: "))
menu(no1)(no2)
```

```
Enter a number: 1
Enter another number: 4
From menu
From sum
Sum:  5
```

20. Write a Python program to detect the number of local variables declared in a function.

```
'''
Sample Output:
3
'''
def locals(argstr):
  locstr=argstr
  return locals.__code__.co_nlocals
inpstr="welcome to cdac mumbai"
print(locals(inpstr))
```

```
2
```

```

```

✓ 2s completed at 10:17 PM