1. Multiplication of Number

```
def multiplicationOfNumber():
  number = int(input("Enter multiplication Number: "))
  if isinstance(number, int):
    for i in range(1, 11):
      print("{} x {} = {}".format(number, i, number*i))
  else:
    print("Accepts only integer")

multiplicationOfNumber()
```

```
Enter multiplication Number: 5
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50
```

2. Print twin primes upto 1000

```
primeNumbers = list()
for p in range(2, 1001):
  for i in range(2, p):
    if p%i == 0:
      break
  else:
    primeNumbers.append(p)
for twin in range(0, len(primeNumbers)-1):
  if primeNumbers[twin+1] - primeNumbers[twin] == 2:
    print((primeNumbers[twin], primeNumbers[twin+1]), end=",")
```

```
(3, 5),(5, 7),(11, 13),(17, 19),(29, 31),(41, 43),(59, 61),(71, 73),(101, 103
```

3. Prime Factors of a number

```
prime = list()
factors = list()
number = int(input("Enter Number: "))
for i in range(2, number+1):
  for j in range(2, i):
    if i%j == 0:
      break
  else:
```

```
  else:
    prime.append(i)
for pr in prime:
  if number == 1: break
  while number % pr == 0:
    factors.append(pr)
    number = number / pr
print(','.join([str(a) for a in factors]))
```

```
Enter Number: 9
3,3
```

4. Permutation and Combination Formulae

```
def factorial(f):
  if f==1 or f==0: return 1
  return f*factorial(f-1)

def p(n,r):
  if n > 0 and r > 0 and n >= r:
    nfac = factorial(n)
    rfac = factorial(n-r)
    return nfac/rfac
  else:
    return "n and r > 0 and n >= r"
def c(n,r):
  if n > 0 and r > 0 and n >= r:
    return p(n,r)/factorial(r)
  else:
    return "n and r > 0 and n >= r"

print("permutation ",p(5,3))
print("combination ",c(5,3))
```

```
permutation  60.0
combination  10.0
```

5. Decimal to binary number.

```
def dtob(n):
  if n > 1:
    dtob(n//2)
  print(n%2, end="")
dtob(17)
```

```
10001
```

6. whether is an Armstrong number

```
def cubesum(n):
  add = 0
  sn = str(n)
  sp = [s for s in sn]
  for p in sp:
    ip = int(p)
    add += ip**3
  return add

def PrintArmstrong(b):
  try:
     if int(b):
       return cubesum(b)
  except:
    return "Only integer accepted"

def isArmstrong(a):
  try:
    if int(a):
      return cubesum(a) == int(a)
  except:
    return "Only integer accepted"

print(PrintArmstrong("153"))
print(isArmstrong("153"))
```

```
153
True
```

7. Product of digits of a number

```
def prodDigits(num):
  try:
    pro = 1
    snum = str(num)
    spnum = [a for a in snum]
    for n in spnum:
      i = int(n)
      pro *= i
    return pro
  except:
    return "Please enter a valid and positive number"

print(prodDigits(566))
```

```
180
```

8. input a number and return its multiplicative digital root and multiplicative persistence respectively

```
def MDR():
  try:
    number = int(input("Enter number: "))
    while number >= 10:
      number = prodDigits(number)
    return number
  except:
    return "Please enter a valid number"

def MPersistence():
  try:
    number = int(input("Enter number: "))
    step = 0
    while number >= 10:
      number = prodDigits(number)
      step += 1
    return step
  except:
    return "Please enter a valid number"

print("MDR: ",MDR())
print("MPersistence: ",MPersistence())
```

```
Enter number: 86
MDR:  6
Enter number: 86
MPersistence:  3
```

9. Find the sum of proper divisors

```
def sumPdivisors(num):
  try:
    # num = int(input("Enter a number: "))
    add = 0
    if num == 0: return "Number should be greater then 0"
    for i in range(1, num):
      if num % i == 0:
        add += i
        # print(i, end=", ")
    return add
  except:
    return "Please enter a valid number."

print(sumPdivisors(284))
```

```
220
```

10. All the perfect number in a given range.

```
def perfect(frm, to):
  if isinstance(frm, int) and isinstance(to, int) and frm < to:
    for i in range(frm, to+1):
      if sumPdivisors(i) == i:
        print(i, end=", ")
      print(i, end=", ")
perfect(2, 100)
```

```
    6, 28,
```

11. Print amicable number in a range.

```
def amicable(frm, to):
  if isinstance(frm, int) and isinstance(to, int) and frm < to:
    for i in range(frm, to+1):
      for j in range(i+1, to+1):
        if sumPdivisors(i) == j and i == sumPdivisors(j):
          print((i, j), end=", ")
amicable(1, 1000)
```

```
    (220, 284),
```

12. filter odd from a list using filter.

```
lst = list(range(1, 100))
flst = list(filter(lambda num: num % 2 != 0, lst))
print(flst)
```

```
    [1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39, 4
```

13. Cube of map() in a given list

```
cubeoflst = list(range(3, 28))
# def cub(x):
  # if x**3 in cubeoflst:
  #   return x
clst = list(map(lambda x: x**3, cubeoflst))
# fclst = list(filter(lambda x: x != None, clst))
print(clst)
```

```
    [27, 64, 125, 216, 343, 512, 729, 1000, 1331, 1728, 2197, 2744, 3375, 4096, 4
```

14. Cube of even numbers in a given list.

```
cubeofevenlst = list(range(3, 47))
flst = list(filter(lambda x: x % 2 == 0, cubeofevenlst))
mlst = list(map(lambda y: y**3, flst))
print(mlst)
```

```
    [64, 216, 512, 1000, 1728, 2744, 4096, 5832, 8000, 10648, 13824, 17576, 21952
```