1. GCD
2. Palindrome
3. Exchange of two variables
4. Circulate the values of n variables
5. Distance between two points
6. Celsius to Fahrenheit and vice-versa
7. Lamda function
8. modules and functions
9. tuple illustration
10. Recursive function
11. global vs local
12. default vs keyword arguments
13. map, filter, reduce
14. type casting vs type coersion

**1. GCD**

**using iteration**

def gcd(a, b):

while b != 0:

a, b = b, a % b

return a

**using recursion**

def gcd(a, b):

if b == 0:

return a

return gcd(b, a % b)

**2. Palindrome number**

**using iteration**

def palindrome(number):

reverse = 0

given = number

while number > 0:

reverse = reverse \* 10 + number % 10

number = number // 10

return given == reverse

**3. Exchange two values**

a = 10

b = 20

temp = b

b = a

a = temp

print(a, b)

(20, 10)

**4. Circulate N values**

N = [2, 3, 4, 5, 6]

last = len(N)-1

temp = N[last]

for index in range(last, 1, -1):

N[index] = N[index-1]

N[0] = temp

print(N)

[6, 3, 3, 4, 5]

**5. Distance between two points**

from math import sqrt

(X1, Y1) = (1, 2)

(X2, Y2) = (4, 6)

Distance = sqrt((X2 -X1)\*\*2 + (Y2- Y1)\*\*2)

print(Distance)

5.0

**6. Celsius to Fahrenheit and vice-versa**

**Fahrenheit to Celsius**

C = 35

F = 9/5\*C + 32

print(F)

95.0

**Celsius to Fahrenheit**

[https://camo.githubusercontent.com/524bc2d3526f2ca3643a1537d7f4ac8b9240bbb4/68747470733a2f2f6c617465782e636f6465636f67732e636f6d2f6769662e6c617465783f432673706163653b3d2673706163653b28462d3332295c667261637b357d7b397d](https://www.codecogs.com/eqnedit.php?latex=C&space;=&space;(F-32)\frac%7b5%7d%7b9%7d)

F = 95.0

C = (F-32)\*(5/9)

print(C)

35.0

**7. Lamda Function**

mul = lambda x,y: x\*y

print(mul(3,4))

12

**13. map, filter, reduce**

**filter**

N = [2, 3, 4, 7, 8, 10]

even = lambda x: x % 2 == 0

result = list(filter(even, N))

print(result)

[2, 4, 8, 10]

**map**

N = [2, 3, 4, 7, 8, 10]

double = lambda x: x\*2

result = list(map(double, N))

[4, 6, 8, 14, 16, 20]

**reduce**

N = [2, 3, 4, 7, 8, 10]

add = lambda x,y: x+y

from functools import reduce

result = reduce(add, N)

print(result)

34

**8. modules and functions**

import math

dir(math) # Functions in the math module

['acos', 'acosh', 'asin', 'asinh', 'atan', 'atan2', 'atanh', 'ceil', 'copysign', 'cos', 'cosh', 'degrees', 'e', 'erf', 'erfc', 'exp', 'expm1', 'fabs', 'factorial', 'floor', 'fmod', 'frexp', 'fsum', 'gamma', 'gcd', 'hypot', 'inf', 'isclose', 'isfinite', 'isinf', 'isnan', 'ldexp', 'lgamma', 'log', 'log10', 'log1p', 'log2', 'modf', 'nan', 'pi', 'pow', 'radians', 'sin', 'sinh', 'sqrt', 'tan', 'tanh', 'tau', 'trunc']

math.sqrt(2)

1.4142135623730951

math.gcd(12,18)

6

**9. tuple illustration**

T = (2, 3, 4, 5, 6)

max(T) # maximum in T 6

min(T) # minimum in T 2

A = T # duplicate T

T[1:4] # (3, 4, 5)

L = list(T) # convert tuple to a list

[2, 3, 4, 5, 6]

**10. Recursive function**

def gcd(a, b):

if b == 0:

return a

return gcd(b, a % b)

**11. global vs local**

a = 10 # global

def add20(num):

a = 20 # local

return num + a

def add10(num):

global a

return num + a

print(add20(5))

25

print(add10(5))

15