# Python Homework05

April 6, 2018

### 0.0.1 Define a function *evalQuadratic(a, b, c, x)*, and return the result of $ax^2 + bx + c$.

```
In [1]: from math import *

        def evalQuadratic(a, b, c, x):
            return a * x ** 2 + b * x + c

        if __name__ == '__main__':
            a = float(input("Please input the val of coef a: "))
            b = float(input("Please input the val of coef b: "))
            c = float(input("Please input the val of coef c: "))
            x = float(input("Please input the val of X: "))
            result = evalQuadratic(a, b, c, x)
            print("The result of {}x^2 + {}x + {} = {}".format(a, b, c, result))
```

```
Please input the val of coef a: 4
Please input the val of coef b: 5
Please input the val of coef c: 6
Please input the val of X: 2
The result of 4.0x^2 + 5.0x + 6.0 = 32.0
```

### 0.0.2 Define a function *isprime(x)*, and find the prime between 1 to 100.

```
In [4]: def isprime(x):
            if x < 2:
                return False
            else:
                for i in range(2, round(sqrt(x) + 1)):
                    if x % i == 0:
                        return False
                return True

        print("The primes in (1, 100) are: ")
        for i in range(1, 100):
            if isprime(i):
                print(i, end = ' ')
```

```
The primes in (1, 100) are:
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97
```

### 0.0.3 Define a function *gcdIter(a, b)* and a function *gcdRecur(a, b)* to find the GCD of two positive integers.

```
In [5]: def gcdIter(a, b):
            maxNum = a if a > b else b
            minNum = b if a > b else a
            while (maxNum % minNum) != 0:
                if (maxNum % minNum) > minNum:
                    maxNum = maxNum % minNum
                else:
                    tempNum = maxNum % minNum
                    maxNum = minNum
                    minNum = tempNum
            return minNum

        def gcdRecur(a, b):
            maxNum = a if a > b else b
            minNum = b if a > b else a
            if maxNum % minNum == 0:
                return minNum
            else:
                return gcdRecur(maxNum % minNum, minNum)
        print("gcd(2,12) = ",gcdIter(2,12))
        print("gcd(6,12) = ",gcdRecur(6,12))
        print("gcd(9,12) = ",gcdRecur(9,12))
        print("gcd(17,12) = ",gcdIter(17,12))
```

```
gcd(2,12) =  2
gcd(6,12) =  6
gcd(9,12) =  3
gcd(17,12) =  1
```

### 0.0.4 Create a module *calculator.py* to calculate the result of add, sub, mult, div of two number; Then create a main function *exec.py* to import the module.

```
In [6]: import calculator
        from functools import reduce
        print('25 + 56 = {}'.format(calculator.add(25, 56)))
        print('86 - 68 = {}'.format(calculator.sub(86, 68)))
        print('50 * 60 = {}'.format(calculator.multi(50, 60)))
        print('99 / 25 = {}'.format(calculator.div(99, 25)))
```

```
25 + 56 = 81
86 - 68 = 18
50 * 60 = 3000
```

```
99 / 25 = 3.96
```

```
In [ ]: # The calculator.py
        def add(num1, num2):
            return num1 + num2

        def sub(num1, num2):
            return num1 - num2

        def multi(num1, num2):
            return num1 * num2

        def div(num1, num2):
            return num1 / num2
```

### 0.0.5 Use the function *filter* to find the number that can be divided by 3 and 5; Use the function *reduce* to calculate the sum of odd number in 100; Use the function *map* to calcutor the cubic of even number in 100.

```
In [14]: def canBeDived(x):
             if x % 3 == 0 and x % 5 == 0:
                 return True
         print(list(filter(canBeDived,range(0,100))))
         print()
         #***********************************************
         def sumODD(x, y):
             if y % 2 != 0:
                 return x + y
             return x

         print(reduce(sumODD,range(1,100)))
         print()
         #***********************************************
         def power3(x):
             return x ** 3

         temp = list(map(power3, range(0,101,2)))
         for i in range(0, len(temp)-1, len(temp)//10):
             print(temp[i:i+11])
```

```
[0, 15, 30, 45, 60, 75, 90]

2500

[0, 8, 64, 216, 512, 1000, 1728, 2744, 4096, 5832, 8000]
[1000, 1728, 2744, 4096, 5832, 8000, 10648, 13824, 17576, 21952, 27000]
[8000, 10648, 13824, 17576, 21952, 27000, 32768, 39304, 46656, 54872, 64000]
```

```
[27000, 32768, 39304, 46656, 54872, 64000, 74088, 85184, 97336, 110592, 125000]
[64000, 74088, 85184, 97336, 110592, 125000, 140608, 157464, 175616, 195112, 216000]
[125000, 140608, 157464, 175616, 195112, 216000, 238328, 262144, 287496, 314432, 343000]
[216000, 238328, 262144, 287496, 314432, 343000, 373248, 405224, 438976, 474552, 512000]
[343000, 373248, 405224, 438976, 474552, 512000, 551368, 592704, 636056, 681472, 729000]
[512000, 551368, 592704, 636056, 681472, 729000, 778688, 830584, 884736, 941192, 1000000]
[729000, 778688, 830584, 884736, 941192, 1000000]
```