

Write a function `printNumbers()` that will print out the numbers from 60 to 48 backwards all on one line with the help of only one variable. Print the messages "Starting" and "Done" before and after the line of numbers. Test the function from `main()`.

In [3]:

```
def printNumbers():
    print("starting")
    for i in range(60,47, -1):
        print(i, end = ' ')

    print("\nDone")

printNumbers()
```

```
starting
60 59 58 57 56 55 54 53 52 51 50 49 48
Done
```

Write a function which takes a single integer parameter, and determines whether it's an even or an odd number. The return type of function should be boolean.

In [5]:

```
def check(value):
    if value % 2 == 0:
        return True
    else:
        return False

val = int(input("Enter a value"))

if check(val) == True:
    print("Even")
else:
    print("Odd")
```

```
Enter a value10
Even
```

Suppose after completing your degree you have been appointed as a programmer at the state of Bella in United Kingdom. The basic currency unit at Bella is coin and the coinage is the quantity of money. The finance minister of Bella has asked you to develop software with following requirements.

1. The program consists of a module that will reads in total number of coins.
2. The programs consist of another module that accepts money as input and convert it into Thousands, Hundreds, Fifties, Tens and Fives.
3. The third module must display the total number of thousands, hundreds, fifties, tens and fives.

In [6]:

```
def data():
    money = int(input("Enter no of Coins : "))
    return money

def calculation(money):
    th = int(money / 1000)

    rem1 = (money - (th * 1000))
    hun = int(rem1 / 100)

    rem2 = (rem1 - (hun * 100))
    fifty = int(rem2 / 50)

    rem3 = (rem2 - (fifty * 50))
    tens = int(rem3 / 10)

    rem4 = (rem3 - (tens * 10))
    fives = int(rem4 / 5)

    coin = rem4 - (fives*5)

    display(money, th, hun, fifty, tens, fives, coin)

def display(m,t,h,f,te,fi,c):
    print ("Ammount ", m, " is divided as : ")
    print ("\t Thousands = ", t)
    print ("\t Hundreds = ", h)
    print ("\t Fifties = ", f)
    print ("\t Tens = ", te)
    print ("\t Fives = ", fi)
    print ("\t Coins = ", c)

val = data()
calculation(val)
```

```
Enter no of Coins : 2257
Ammount 2257 is divided as :
    Thousands = 2
    Hundreds = 2
    Fifties = 1
    Tens = 0
    Fives = 1
    Coins = 2
```

Write a function "estimateCost(...)" to gauge the expected cost of an item in a specified number of years. The program asks for the cost of the item, the number of years from now that the item will be purchased, the rate of inflation and make a call to "estimateCost (...)" which will estimate and return the cost of the item after the specified period. If user enters the inflation rate as a percentage like 5.6 (%), your program should then convert the percent to

a fraction such as 0.056 and estimate the price adjusted for inflation. Test the functionality of your module from main().

```
In [2]: def estimateCost(cost, years, rate):
        expected = float(cost + (years* (cost * (rate/100))))
        return expected

def start():
    cost = float(input("Enter Cost : "))
    years = int(input("Enter Years : "))
    rate = float(input("Enter Rate : "))

    es = estimateCost(cost, years,rate)
    print("Estimated Cost is : ", es)

start()
```

```
Enter Cost : 1000
Enter Years : 5
Enter Rate : 5.6
Estimated Cost is : 1280.0
```

Write a function that will print Fibonacci series up to n number, the n is entered by a user and passed to a function as an argument. By definition, the first two Fibonacci numbers are 0 and 1, and each remaining number is the sum of the previous two.

$F_n = F_{n-1} + F_{n-2}$ The Fibonacci series is 0, 1, 1, 2, 3, 5, 8, 13, 21...

```
In [12]: def output(n):
        print(n, end = '\t')

def fibonacci(n):
    first ,second,term = 0, 1, 0
    if n == 0:
        print(first)
    else:
        print(first,"\t",second, end = '\t')

    for i in range(2,n+1):
        term = first + second
        output(term)
        first = second
        second = term

n = int(input("Enter n : "))
fibonacci(n)
```

Enter n : 15

0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610

In [1]:

```
def extract(data):
    print("Name is : ", data[0:17])
    print("Extension is : ", data[18: data.find(' ')])

extract("bitf19m526.myasir@pucit.edu.pk jan 1 2021")
```

Name is : bitf19m526.myasir

Extension is : pucit.edu.pk

Functions in which number of args are unknown

In [7]:

```
def sum(a, *b):
    total = a;
    for i in b:
        total += i

    print(total)

sum(3,4,5,6,7) # a = 3, all others will be mapped as a list to b

l1 = [4,5,6,7]
sum(3, *l1) #So a list by value can also be sent to b
```

25

25

Write a Function that accept arbitrary keywords

In [10]:

```
def add(a, **kwargs):
    total = a;

    for key, value in kwargs.items():
        print("KEY : ", key, "\t", "VALUE : ", value)
        total += value

    print("Sum = ",total)

add(10, w = 15, y = 20)
```

KEY : w VALUE : 15

KEY : y VALUE : 20

Sum = 45

Write a Function that accepts positional and arbitrary keywords arguments

In [14]:

```
def func(*args, **kwargs):  
    print(args, kwargs)  
  
func(10,20,30, w = 15, y = 20)
```

```
(10, 20, 30) {'w': 15, 'y': 20}
```