

LAB 08: Quick Sort

CS211 – Data Structures and Algorithms
Usman Institute of Technology
Fall 2019

- **How to submit:**

- Create an account on <http://www.turnitin.com/> as a Student (if you don't have already)
- Use following information at time of sign-up

- CS Section A**

- Class ID: 22664649
 - Enrollment Key: DSFALL19CSA

- CS Section B**

- Class ID: 22664651
 - Enrollment Key: DSFALL19CSB

A. Implement the following functions of recursion in Python.

1. A function **Write()** that takes an argument n and prints the numbers in reverse order recursively.

```
def Write(n):  
    // your code goes here
```

Example:

```
Write(5)
```

```
#The result should be like this:
```

```
5  
4  
3  
2  
1
```

2. Add a function **Factorial()** that returns the factorial of a number.

```
def Factorial(n):  
    // your code goes here
```

Example:

```
Factorial(5)
```

```
#The result should be like this:  
5! = 5x4x3x2x1 = 120
```

3. A function **GCD()** that takes two numbers and returns their greatest common divisor.

```
def GCD(a,b):  
    // your code goes here
```

Example:

```
GCD(8,12)
```

```
#The function should return 4
```

Algorithm:

```
GCD(a,0) = a           Base Case  
GCD(a,b) = GCD(b,a mod b) Recursive case
```

4. A function **BinarySearch()** that implements the binary search algorithm for non-empty sorted array using recursion. The function should take the arguments List, value, low, high and returns the location of the searched value.

```
def BinarySearch(List,low,high,value):  
    // your code goes here
```

5. A function **QuickSort()** which sorts the list in ascending/descending order.

```
def QuickSort():  
    // your code goes here
```

```
function partitionFunc(left, right, pivot)  
    leftPointer = left  
    rightPointer = right - 1  
  
    while True do  
        while A[++leftPointer] < pivot do  
            //do-nothing  
        end while  
  
        while rightPointer > 0 && A[--rightPointer] > pivot do  
            //do-nothing  
        end while
```

```
        if leftPointer >= rightPointer
            break
        else
            swap leftPointer, rightPointer
        end if

    end while

    swap leftPointer, right
    return leftPointer

end function

procedure quickSort(left, right)

    if right-left <= 0
        return
    else
        pivot = A[right]
        partition = partitionFunc(left, right, pivot)
        quickSort(left, partition-1)
        quickSort(partition+1, right)
    end if

end procedure

Source: TutorialsPoint
```