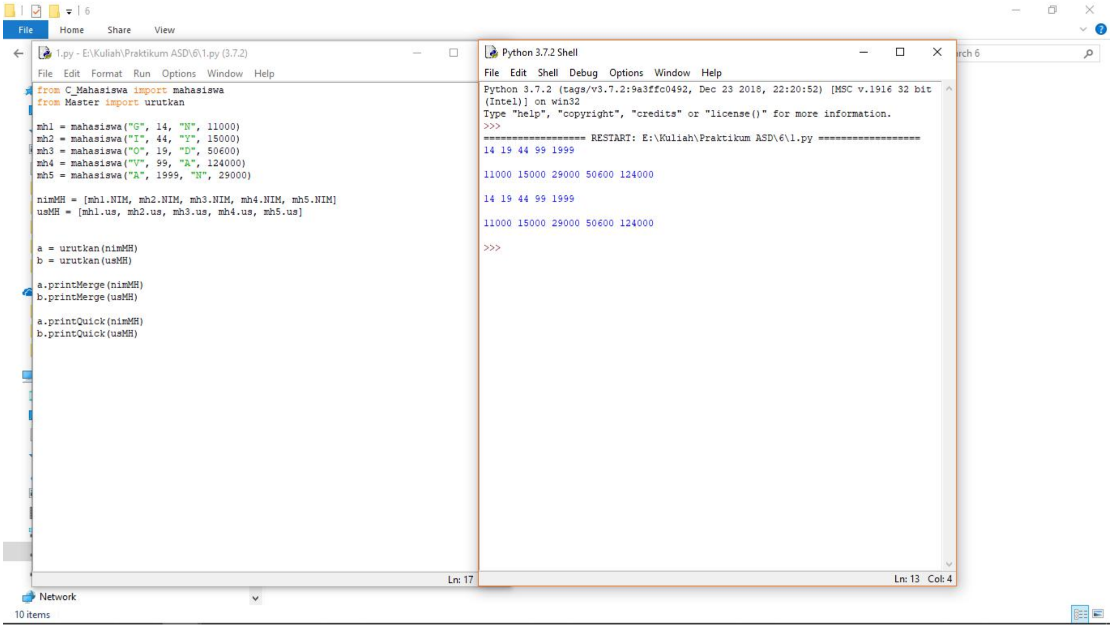
Nama : Giovanny Dani Saputra

NIM : L200170159

1.



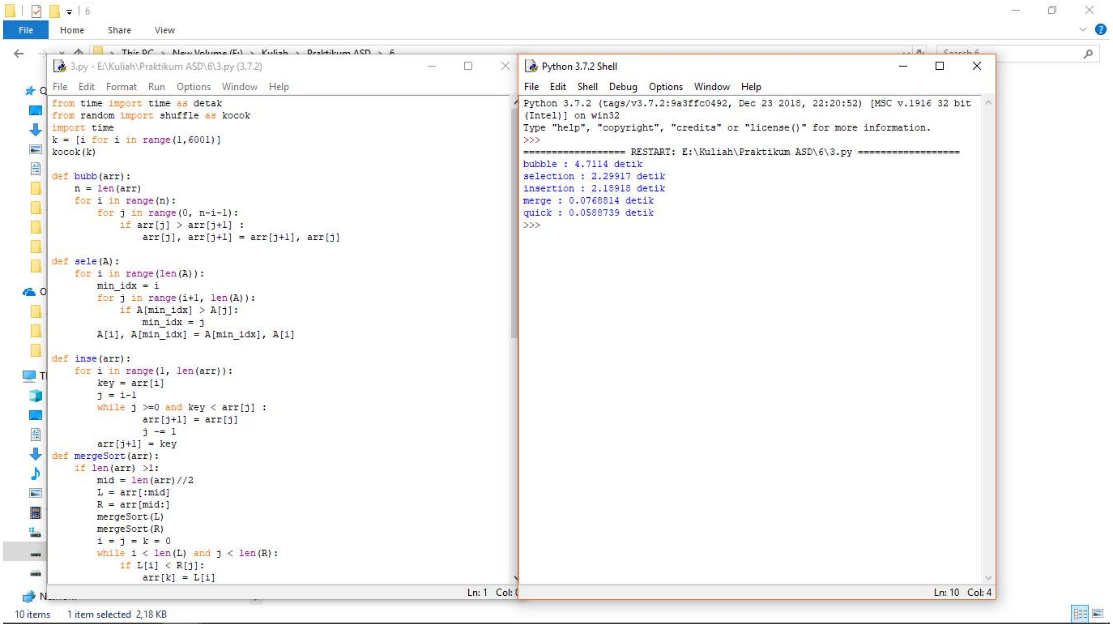
6
File
Home
Share
View
1.py - E:\Kuliah\Praktikum ASD\6\1.py (3.7.2)
File Edit Format Run Options Window Help
Ln: 17
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Ln: 13 Col: 4
Network
10 items

```
from C_Mahasiswa import mahasiswa
from Master import urutkan

mh1 = mahasiswa("G", 14, "N", 11000)
mh2 = mahasiswa("I", 44, "Y", 15000)
mh3 = mahasiswa("O", 19, "D", 50600)
mh4 = mahasiswa("V", 99, "A", 124000)
mh5 = mahasiswa("A", 1999, "N", 29000)

nimMH = [mh1.NIM, mh2.NIM, mh3.NIM, mh4.NIM, mh5.NIM]
usMH = [mh1.us, mh2.us, mh3.us, mh4.us, mh5.us]


a = urutkan(nimMH)
b = urutkan(usMH)

a.printMerge(nimMH)
b.printMerge(usMH)

a.printQuick(nimMH)
b.printQuick(usMH)
```

```
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:20:52) [MSC v.1916 32 bit
(Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
================== RESTART: E:\Kuliah\Praktikum ASD\6\1.py ==================
14 19 44 99 1999

11000 15000 29000 50600 124000

14 19 44 99 1999

11000 15000 29000 50600 124000

>>>
```

2. -

3.

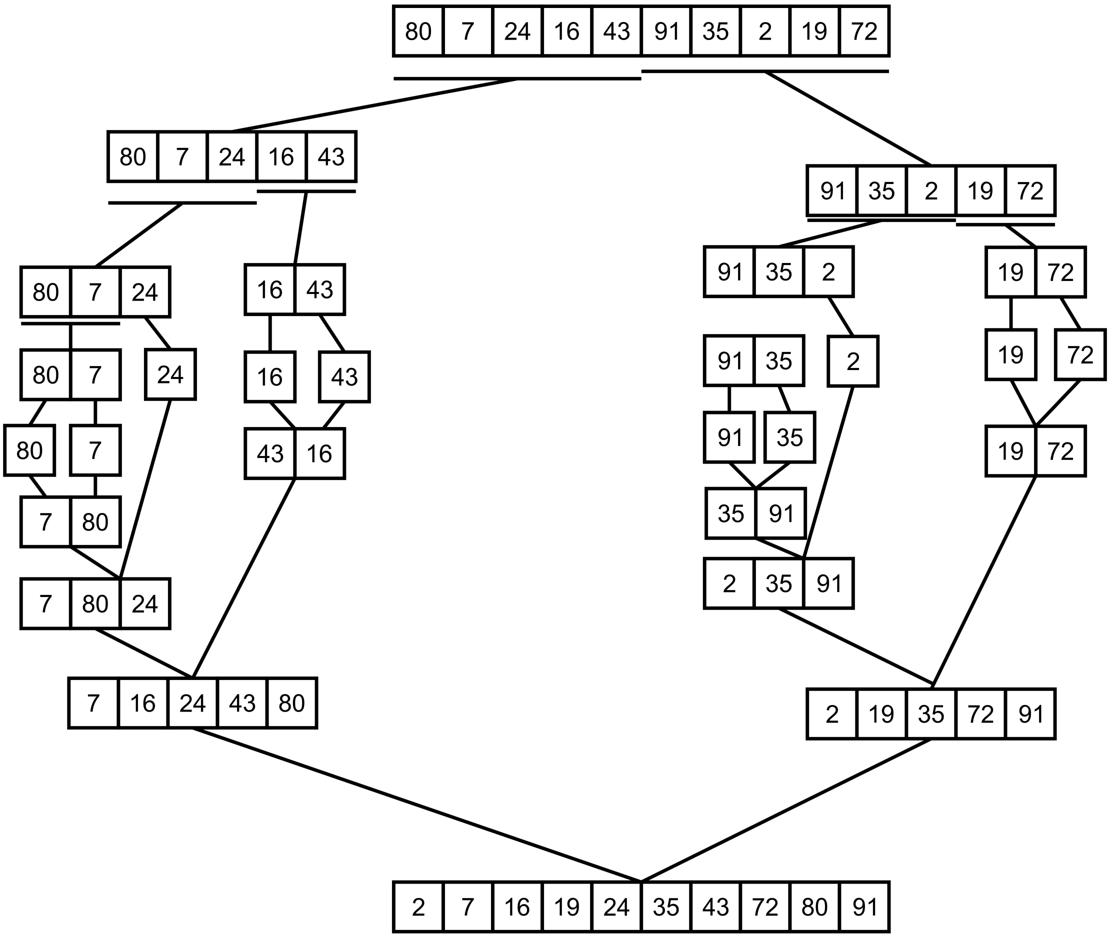


6
File
Home
Share
View
This PC > New Volume (E:) > Kuliah > Praktikum ASD > 6
Search 6
3.py - E:\Kuliah\Praktikum ASD\6\3.py (3.7.2)
File Edit Format Run Options Window Help
Ln: 1 Col: (
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Ln: 10 Col: 4
10 items 1 item selected 2,18 KB

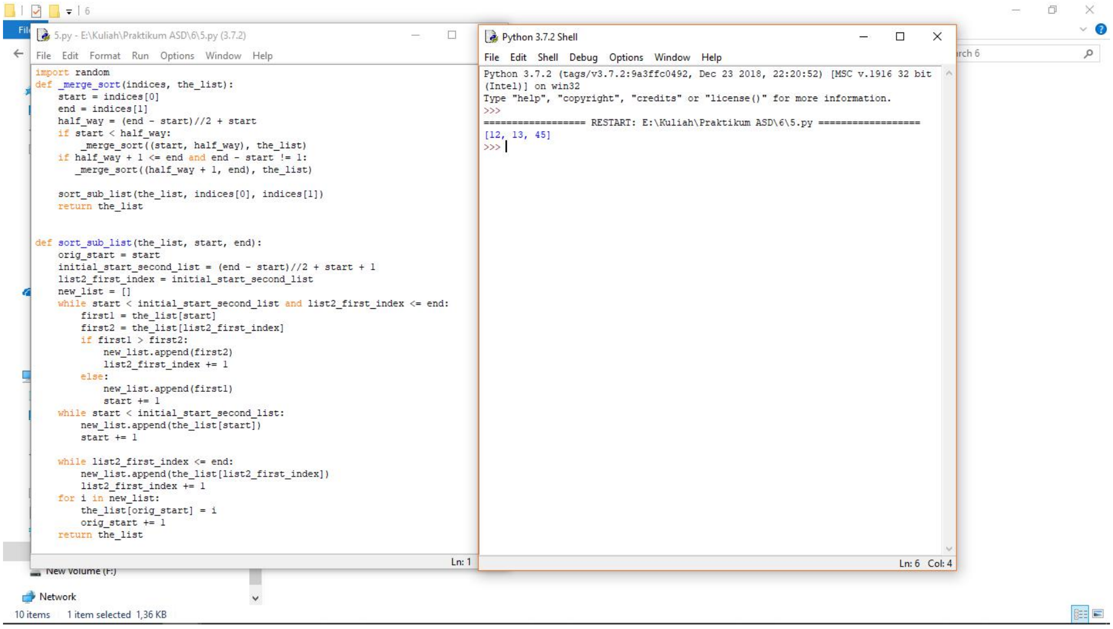
```
from time import time as detak
from random import shuffle as kocok
import time
k = [i for i in range(1,6001)]
kocok(k)

def bubb(arr):
    n = len(arr)
    for i in range(n):
        for j in range(0, n-i-1):
            if arr[j] > arr[j+1] :
                arr[j], arr[j+1] = arr[j+1], arr[j]

def sele(A):
    for i in range(len(A)):
        min_idx = i
        for j in range(i+1, len(A)):
            if A[min_idx] > A[j]:
                min_idx = j
        A[i], A[min_idx] = A[min_idx], A[i]

def inse(arr):
    for i in range(1, len(arr)):
        key = arr[i]
        j = i-1
        while j >=0 and key < arr[j] :
                arr[j+1] = arr[j]
                j -= 1
        arr[j+1] = key
def mergeSort(arr):
    if len(arr) >1:
        mid = len(arr)//2
        L = arr[:mid]
        R = arr[mid:]
        mergeSort(L)
        mergeSort(R)
        i = j = k = 0
        while i < len(L) and j < len(R):
            if L[i] < R[j]:
                arr[k] = L[i]
```

```
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:20:52) [MSC v.1916 32 bit
(Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
================== RESTART: E:\Kuliah\Praktikum ASD\6\3.py ==================
bubble : 4.7114 detik
selection : 2.29917 detik
insertion : 2.18918 detik
merge : 0.0768814 detik
quick : 0.0588739 detik
>>>
```

4 A.


80 7 24 16 43 91 35 2 19 72
80 7 24 16 43
91 35 2 19 72
80 7 24
16 43
91 35 2
19 72
80 7
24
16
43
91 35
2
19
72
80
7
43 16
91
35
19 72
7 80
35 91
7 80 24
2 35 91
7 16 24 43 80
2 19 35 72 91
2 7 16 19 24 35 43 72 80 91

5.

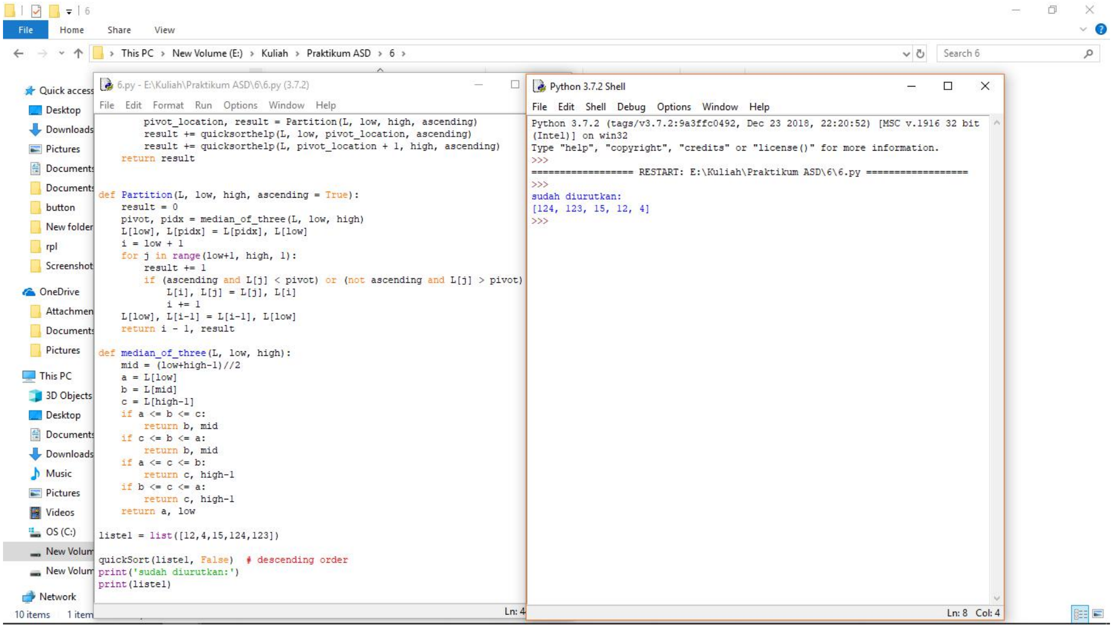


5.py - E:\Kuliah\Praktikum ASD\6\5.py (3.7.2)
File Edit Format Run Options Window Help
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Ln: 1
Ln: 6 Col: 4
New Volume (F:)
Network
10 items 1 item selected 1,36 KB

```
import random
def _merge_sort(indices, the_list):
    start = indices[0]
    end = indices[1]
    half_way = (end - start)//2 + start
    if start < half_way:
        _merge_sort((start, half_way), the_list)
    if half_way + 1 <= end and end - start != 1:
       _merge_sort((half_way + 1, end), the_list)

    sort_sub_list(the_list, indices[0], indices[1])
    return the_list


def sort_sub_list(the_list, start, end):
    orig_start = start
    initial_start_second_list = (end - start)//2 + start + 1
    list2_first_index = initial_start_second_list
    new_list = []
    while start < initial_start_second_list and list2_first_index <= end:
        first1 = the_list[start]
        first2 = the_list[list2_first_index]
        if first1 > first2:
            new_list.append(first2)
            list2_first_index += 1
        else:
            new_list.append(first1)
            start += 1
    while start < initial_start_second_list:
        new_list.append(the_list[start])
        start += 1

    while list2_first_index <= end:
        new_list.append(the_list[list2_first_index])
        list2_first_index += 1
    for i in new_list:
        the_list[orig_start] = i
        orig_start += 1
    return the_list
```

```
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:20:52) [MSC v.1916 32 bit
(Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
================= RESTART: E:\Kuliah\Praktikum ASD\6\5.py =================
[12, 13, 45]
>>>
```

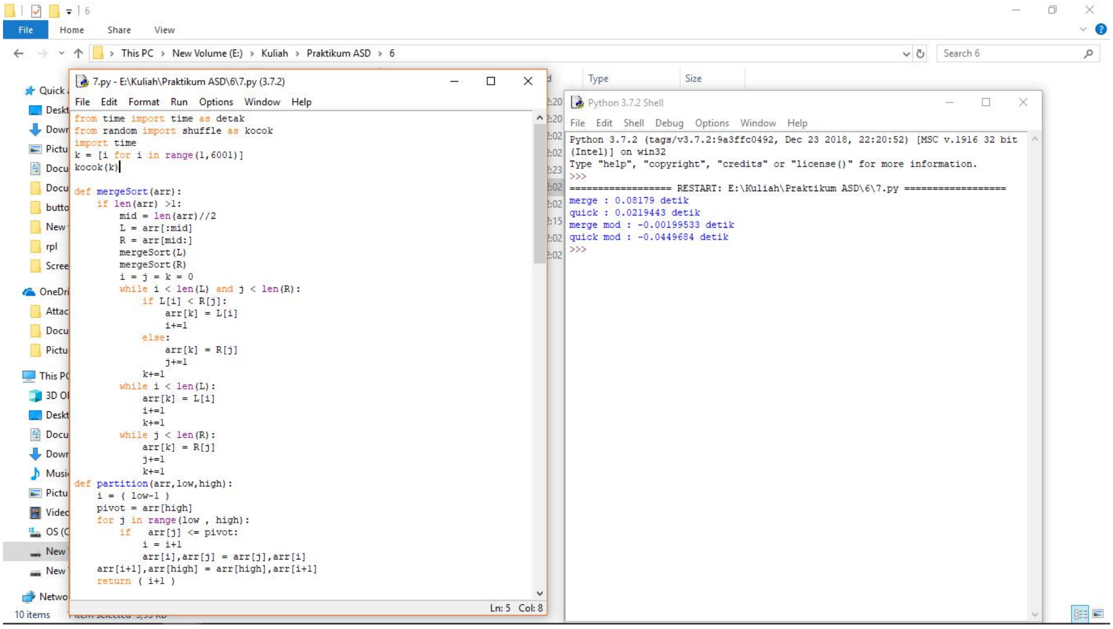
6.



```
        pivot_location, result = Partition(L, low, high, ascending)
        result += quicksorthelp(L, low, pivot_location, ascending)
        result += quicksorthelp(L, pivot_location + 1, high, ascending)
    return result


def Partition(L, low, high, ascending = True):
    result = 0
    pivot, pidx = median_of_three(L, low, high)
    L[low], L[pidx] = L[pidx], L[low]
    i = low + 1
    for j in range(low+1, high, 1):
        result += 1
        if (ascending and L[j] < pivot) or (not ascending and L[j] > pivot)
            L[i], L[j] = L[j], L[i]
            i += 1
    L[low], L[i-1] = L[i-1], L[low]
    return i - 1, result

def median_of_three(L, low, high):
    mid = (low+high-1)//2
    a = L[low]
    b = L[mid]
    c = L[high-1]
    if a <= b <= c:
        return b, mid
    if c <= b <= a:
        return b, mid
    if a <= c <= b:
        return c, high-1
    if b <= c <= a:
        return c, high-1
    return a, low

liste1 = list([12,4,15,124,123])

quickSort(liste1, False)  # descending order
print('sudah diurutkan:')
print(liste1)
```

```
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:20:52) [MSC v.1916 32 bit
(Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
================= RESTART: E:\Kuliah\Praktikum ASD\6\6.py =================
>>>
sudah diurutkan:
[124, 123, 15, 12, 4]
>>>
```

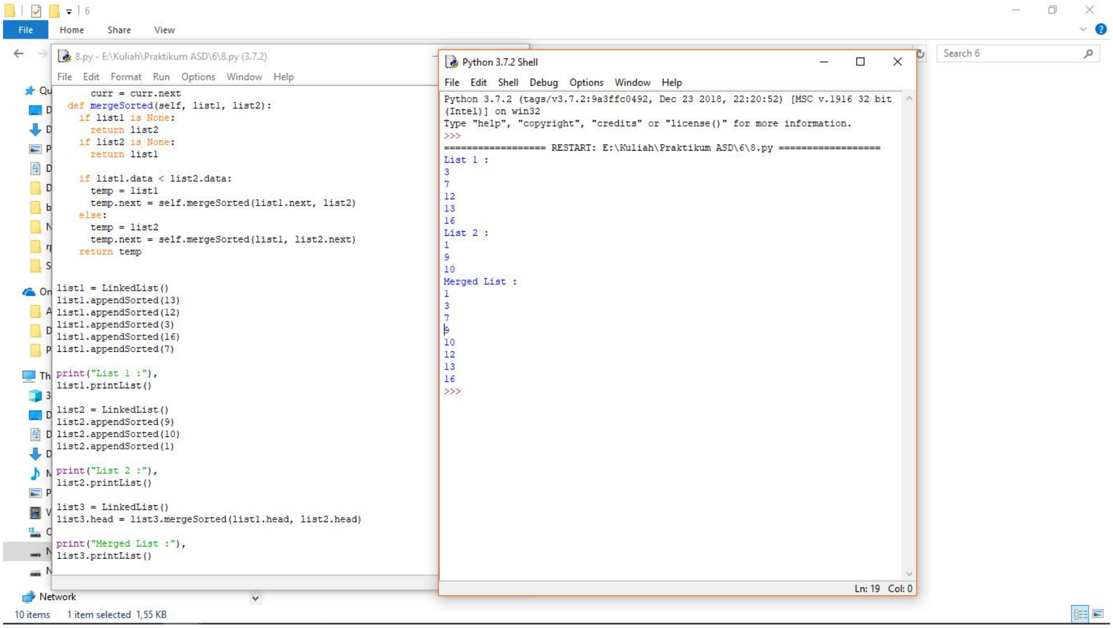
7.


This PC › New Volume (E:) › Kuliah › Praktikum ASD › 6
7.py - E:\Kuliah\Praktikum ASD\6\7.py (3.7.2)
File Edit Format Run Options Window Help
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Ln: 5 Col: 8
10 items

```
from time import time as detak
from random import shuffle as kocok
import time
k = [i for i in range(1,6001)]
kocok(k)

def mergeSort(arr):
    if len(arr) >1:
        mid = len(arr)//2
        L = arr[:mid]
        R = arr[mid:]
        mergeSort(L)
        mergeSort(R)
        i = j = k = 0
        while i < len(L) and j < len(R):
            if L[i] < R[j]:
                arr[k] = L[i]
                i+=1
            else:
                arr[k] = R[j]
                j+=1
            k+=1
        while i < len(L):
            arr[k] = L[i]
            i+=1
            k+=1
        while j < len(R):
            arr[k] = R[j]
            j+=1
            k+=1
def partition(arr,low,high):
    i = ( low-1 )
    pivot = arr[high]
    for j in range(low , high):
        if   arr[j] <= pivot:
            i = i+1
            arr[i],arr[j] = arr[j],arr[i]
    arr[i+1],arr[high] = arr[high],arr[i+1]
    return ( i+1 )
```

```
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:20:52) [MSC v.1916 32 bit
(Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
================== RESTART: E:\Kuliah\Praktikum ASD\6\7.py ==================
merge : 0.08179 detik
quick : 0.0219443 detik
merge mod : -0.00199533 detik
quick mod : -0.0449684 detik
>>>
```

8.


8.py - E:\Kuliah\Praktikum ASD\6\8.py (3.7.2)
File Edit Format Run Options Window Help
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Ln: 19 Col: 0
Network
10 items 1 item selected 1,55 KB

```
      curr = curr.next
  def mergeSorted(self, list1, list2):
    if list1 is None:
      return list2
    if list2 is None:
      return list1

    if list1.data < list2.data:
      temp = list1
      temp.next = self.mergeSorted(list1.next, list2)
    else:
      temp = list2
      temp.next = self.mergeSorted(list1, list2.next)
    return temp


list1 = LinkedList()
list1.appendSorted(13)
list1.appendSorted(12)
list1.appendSorted(3)
list1.appendSorted(16)
list1.appendSorted(7)

print("List 1 :"),
list1.printList()

list2 = LinkedList()
list2.appendSorted(9)
list2.appendSorted(10)
list2.appendSorted(1)

print("List 2 :"),
list2.printList()

list3 = LinkedList()
list3.head = list3.mergeSorted(list1.head, list2.head)

print("Merged List :"),
list3.printList()
```

```
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:20:52) [MSC v.1916 32 bit
(Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
================== RESTART: E:\Kuliah\Praktikum ASD\6\8.py ==================
List 1 :
3
7
12
13
16
List 2 :
1
9
10
Merged List :
1
3
7
9
10
12
13
16
>>>
```