

```
110_Modul6.py - C:\Users\Ramadhanu\Documents\Semester 4\110_Modul6.py (3.8.1)
File Edit Format Run Options Window Help

#Nomor 1
class Mahasiswa(object):
    def __init__(self, nama, NIM, kota, us):
        self.nama = nama
        self.NIM = NIM
        self.kotaTinggal = kota
        self.uangSaku = us

a0 = Mahasiswa('Ika', 10, 'Sukoharjo', 240000)
a1 = Mahasiswa('Budi', 51, 'Sragen', 230000)
a2 = Mahasiswa('Ahmad', 2, 'Surakarta', 250000)
a3 = Mahasiswa('Chandra', 18, 'Surakarta', 235000)
a4 = Mahasiswa('Eka', 4, 'Boyolali', 240000)
a5 = Mahasiswa('Fandi', 31, 'Salatiga', 250000)
a6 = Mahasiswa('Deni', 13, 'Klaten', 245000)
a7 = Mahasiswa('Galuh', 5, 'Wonogiri', 245000)
a8 = Mahasiswa('Janto', 23, 'Klaten', 245000)
a9 = Mahasiswa('Hanan', 64, 'Karanganyar', 270000)
a10 = Mahasiswa('Khalid', 29, 'Purwodadi', 230000)

Daftar = [a0, a1, a2, a3, a4, a5, a6, a7, a8, a9, a10]

def urutkanNIM(a):
    baru = []
    for i in range(len(a)):
        baru[a[i].nama] = a[i].NIM
    listofTuples = sorted(baru.items(), key = lambda x: x[1])
    for elem in listofTuples:
        print(elem[0], ': ', elem[1])

urutkanNIM(Daftar)

#Nomor 2
def bubblesort(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]
    return arr

def gabung(a,b):
    c = []
    c = a+b
    n = len(c)
    for i in range(n):
        for j in range(0, n-i-1):

Python 3.8.1 Shell
File Edit Shell Debug Options Window Help

>>>
===== RESTART: C:\Users\Ramadhanu\Documents\Semester 4\110_Modul6.py =====
Ahmad : 2
Eka : 4
Galuh : 5
Ika : 10
Deni : 13
Chandra : 18
Janto : 23
Khalid : 29
Fandi : 31
Budi : 51
Hanan : 64
[2, 5, 6, 10, 12, 32, 45]
[8, 14, 20, 26, 40]
[2, 5, 6, 8, 10, 12, 14, 20, 26, 32, 40, 45]
bubble : 23.7104 detik
selection : 11.0129 detik
insertion : 9.64819 detik
merge : 0.176281 detik
quick : 0.0924933 detik
[12, 13, 45]
sorted : [124, 123, 15, 12, 4]
merge : 0.164328 detik
quick : 0.0969451 detik
merge mod : -0.02494 detik
quick mod : -0.241241 detik
List 1 :
3
7
12
13
16
List 2 :
1
9
10
Merged List :
1
3
7
9
10
12
13
16
>>>

Ln: 32 Col: 0
Ln: 25 Col: 0

Activate Windows
Go to Settings to activate Windows.
```

```
110_Modul6.py - C:\Users\Ramadhanu\Documents\Semester 4\110_Modul6.py (3.8.1)
File Edit Format Run Options Window Help

for i in range(n):
    for j in range(0, n-i-1):
        if c[j] > c[j+1]:
            c[j], c[j+1] = c[j+1], c[j]
    return c

a = [5,45,12,32,6,10,2]
b = [26,8,20,14,40]
a,b = bubblesort(a),bubblesort(b)

print(a)
print(b)
print(gabung(a,b))

#Nomor 3 dan 4
from time import time as detik
from random import shuffle as kocok
import time

k = [1 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 insr(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):

Python 3.8.1 Shell
File Edit Shell Debug Options Window Help

>>>
===== RESTART: C:\Users\Ramadhanu\Documents\Semester 4\110_Modul6.py =====
Ahmad : 2
Eka : 4
Galuh : 5
Ika : 10
Deni : 13
Chandra : 18
Janto : 23
Khalid : 29
Fandi : 31
Budi : 51
Hanan : 64
[2, 5, 6, 10, 12, 32, 45]
[8, 14, 20, 26, 40]
[2, 5, 6, 8, 10, 12, 14, 20, 26, 32, 40, 45]
bubble : 23.7104 detik
selection : 11.0129 detik
insertion : 9.64819 detik
merge : 0.176281 detik
quick : 0.0924933 detik
[12, 13, 45]
sorted : [124, 123, 15, 12, 4]
merge : 0.164328 detik
quick : 0.0969451 detik
merge mod : -0.02494 detik
quick mod : -0.241241 detik
List 1 :
3
7
12
13
16
List 2 :
1
9
10
Merged List :
1
3
7
9
10
12
13
16
>>>

Ln: 32 Col: 0
Ln: 25 Col: 0

Activate Windows
Go to Settings to activate Windows.
```

```
110_Modul6.py - C:\Users\Ramadhanu\Documents\Semester 4\110_Modul6.py (3.8.1)
File Edit Format Run Options Window Help

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 )

def quickSort(arr,low,high):
    if low < high:
        pi = partition(arr,low,high)
        quickSort(arr, low, pi-1)
        quickSort(arr, pi+1, high)

bub = k[:]
sel = k[:]
ins = k[:]
mer = k[:]
qui = k[:]
```

```
Python 3.8.1 Shell
File Edit Shell Debug Options Window Help

>>>
===== RESTART: C:\Users\Ramadhanu\Documents\Semester 4\110_Modul6.py =====
Ahmad : 2
Eka : 4
Galuh : 5
Ika : 10
Denti : 13
Chandra : 18
Janto : 23
Khalid : 29
Fandi : 31
Budi : 51
Hasan : 64
[2, 5, 6, 10, 12, 32, 45]
[8, 14, 20, 26, 40]
[2, 5, 6, 8, 10, 12, 14, 20, 26, 32, 40, 45]
bubble : 23.7104 detik
selection : 11.0129 detik
insertion : 9.64819 detik
merge : 0.176281 detik
quick : 0.0924933 detik
[12, 13, 45]
sorted : [124, 123, 15, 12, 4]
merge : 0.164328 detik
quick : 0.0969451 detik
merge mod : -0.02494 detik
quick mod : -0.241241 detik
List 1 :
3
7
12
13
16
Merged List :
1
9
10
12
13
16
List 2 :
1
9
10
12
13
16
Merged List :
3
7
9
10
12
13
16
>>>
```

Ln: 32 Col: 0

Ln: 25 Col: 0

2024  
04/04/2020

```
110_Modul6.py - C:\Users\Ramadhanu\Documents\Semester 4\110_Modul6.py (3.8.1)
File Edit Format Run Options Window Help

quickSort(arr, pi+1, high)

bub = k[:]
sel = k[:]
ins = k[:]
mer = k[:]
qui = k[:]

aw=detak();bubb(bub);ak=detak();print('bubble : %g detik' %(ak-aw));
aw=detak();sele(sel);ak=detak();print('selection : %g detik' %(ak-aw));
aw=detak();inse(ins);ak=detak();print('insertion : %g detik' %(ak-aw));
aw=detak();mergeSort(mer);ak=detak();print('merge : %g detik' %(ak-aw));
aw=detak();quickSort(qui,0,len(qui)-1);ak=detak();print('quick : %g detik' %(ak-aw));

#Nomor 5
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:
```

```
Python 3.8.1 Shell
File Edit Shell Debug Options Window Help

>>>
===== RESTART: C:\Users\Ramadhanu\Documents\Semester 4\110_Modul6.py =====
Ahmad : 2
Eka : 4
Galuh : 5
Ika : 10
Denti : 13
Chandra : 18
Janto : 23
Khalid : 29
Fandi : 31
Budi : 51
Hasan : 64
[2, 5, 6, 10, 12, 32, 45]
[8, 14, 20, 26, 40]
[2, 5, 6, 8, 10, 12, 14, 20, 26, 32, 40, 45]
bubble : 23.7104 detik
selection : 11.0129 detik
insertion : 9.64819 detik
merge : 0.176281 detik
quick : 0.0924933 detik
[12, 13, 45]
sorted : [124, 123, 15, 12, 4]
merge : 0.164328 detik
quick : 0.0969451 detik
merge mod : -0.02494 detik
quick mod : -0.241241 detik
List 1 :
3
7
12
13
16
Merged List :
1
9
10
12
13
16
List 2 :
1
9
10
12
13
16
Merged List :
3
7
9
10
12
13
16
>>>
```

Ln: 32 Col: 0

Ln: 25 Col: 0

2024  
04/04/2020

```
110_Modul6.py - C:\Users\Ramadhanu\Documents\Semester 4\110_Modul6.py (3.8.1)
File Edit Format Run Options Window Help
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

def merge_sort(the_list):
    return _merge_sort((0, len(the_list) - 1), the_list)

print(merge_sort([13,45,12]))

#Momor 6
def quickSort(L, ascending = True):
    quicksorthelp(L, 0, len(L), ascending)

def quicksorthelp(L, low, high, ascending = True):
    result = 0
    if low < high:
        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

Python 3.8.1 Shell
File Edit Shell Debug Options Window Help
>>>
===== RESTART: C:\Users\Ramadhanu\Documents\Semester 4\110_Modul6.py =====
Ahmad : 2
Eka : 4
Galuh : 5
Ika : 10
Demi : 13
Chandra : 18
Janto : 23
Khalid : 29
Fandi : 31
Budi : 51
Hasan : 64
[2, 5, 6, 10, 12, 32, 45]
[8, 14, 20, 26, 40]
[2, 5, 6, 8, 10, 12, 14, 20, 26, 32, 40, 45]
bubble : 23.7104 detik
selection : 11.0129 detik
insertion : 9.64819 detik
merge : 0.176281 detik
quick : 0.0924933 detik
[12, 13, 45]
sorted : [124, 123, 15, 12, 4]
merge : 0.164328 detik
quick : 0.0969651 detik
insertion : 9.64819 detik
merge mod : -0.02494 detik
quick mod : -0.241241 detik
List 1 :
3
7
12
13
16
Merged List :
1
3
7
9
10
12
13
16
>>>

Ln: 32 Col: 0
Ln: 25 Col: 0
2024
04/04/2020
Activate Windows
Go to Settings to activate Windows.
```

```
110_Modul6.py - C:\Users\Ramadhanu\Documents\Semester 4\110_Modul6.py (3.8.1)
File Edit Format Run Options Window Help
if b <= c <= a:
    return c, high-1
return a, low

listel = list([112,4,15,124,123])
quickSort(listel, False) # descending order
print('sorted :', listel)

#Momor 7
from time import time as detik
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

Python 3.8.1 Shell
File Edit Shell Debug Options Window Help
>>>
===== RESTART: C:\Users\Ramadhanu\Documents\Semester 4\110_Modul6.py =====
Ahmad : 2
Eka : 4
Galuh : 5
Ika : 10
Demi : 13
Chandra : 18
Janto : 23
Khalid : 29
Fandi : 31
Budi : 51
Hasan : 64
[2, 5, 6, 10, 12, 32, 45]
[8, 14, 20, 26, 40]
[2, 5, 6, 8, 10, 12, 14, 20, 26, 32, 40, 45]
bubble : 23.7104 detik
selection : 11.0129 detik
insertion : 9.64819 detik
merge : 0.176281 detik
quick : 0.0924933 detik
[12, 13, 45]
sorted : [124, 123, 15, 12, 4]
merge : 0.164328 detik
quick : 0.0969651 detik
insertion : 9.64819 detik
merge mod : -0.02494 detik
quick mod : -0.241241 detik
List 1 :
3
7
12
13
16
Merged List :
1
3
7
9
10
12
13
16
>>>

Ln: 32 Col: 0
Ln: 25 Col: 0
2025
04/04/2020
Activate Windows
Go to Settings to activate Windows.
```

```
110_Modul6.py - C:\Users\Ramadhanu\Documents\Semester 4\110_Modul6.py (3.8.1)
File Edit Format Run Options Window Help

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)

def quickSort(arr, low, high):
    if low < high:
        pi = partition(arr, low, high)
        quickSort(arr, low, pi-1)
        quickSort(arr, pi+1, high)

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])

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

def merge_sort(the_list):
    return _merge_sort((0, len(the_list) - 1), the_list)

def quickSortMOD(L, ascending = True):
    quickSortHelp(L, 0, len(L), ascending)

def quickSortHelp(L, low, high, ascending = True):
    result = 0
    if low < high:
        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_idx = median_of_three(L, low, high)
    L[low], L[pivot_idx] = L[pivot_idx], 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 <= a <= c:
        return a, low
    if c <= a <= b:
        return c, high-1
    if b <= c <= a:
        return b, mid

Python 3.8.1 Shell
File Edit Shell Debug Options Window Help

>>>
===== RESTART: C:\Users\Ramadhanu\Documents\Semester 4\110_Modul6.py =====
Ahmad : 2
Eka : 4
Galuh : 5
Ika : 10
Demi : 13
Chandra : 18
Janto : 23
Khailid : 29
Fandi : 31
Budi : 51
Hasan : 64
[2, 5, 6, 8, 10, 12, 32, 45]
[8, 14, 20, 26, 40]
[2, 5, 6, 8, 10, 12, 14, 20, 26, 32, 40, 45]
Bubble : 23.7104 detik
selection : 11.0129 detik
insertion : 9.64819 detik
merge : 0.176281 detik
quick : 0.0924933 detik
[12, 13, 45]
sorted : [124, 123, 15, 12, 4]
merge : 0.164328 detik
quick : 0.0969451 detik
merge mod : -0.02494 detik
quick mod : -0.241241 detik
List 1 :
3
7
12
13
16
List 2 :
1
9
10
Merged List :
1
3
7
9
10
12
13
16
>>>

Ln: 32 Col: 0
Ln: 25 Col: 0

Activate Windows
Go to Settings to activate Windows.
```

```
110_Modul6.py - C:\Users\Ramadhanu\Documents\Semester 4\110_Modul6.py (3.8.1)
File Edit Format Run Options Window Help

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

def merge_sort(the_list):
    return _merge_sort((0, len(the_list) - 1), the_list)

def quickSortMOD(L, ascending = True):
    quickSortHelp(L, 0, len(L), ascending)

def quickSortHelp(L, low, high, ascending = True):
    result = 0
    if low < high:
        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_idx = median_of_three(L, low, high)
    L[low], L[pivot_idx] = L[pivot_idx], 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 <= a <= c:
        return a, low
    if c <= a <= b:
        return c, high-1
    if b <= c <= a:
        return b, mid

Python 3.8.1 Shell
File Edit Shell Debug Options Window Help

>>>
===== RESTART: C:\Users\Ramadhanu\Documents\Semester 4\110_Modul6.py =====
Ahmad : 2
Eka : 4
Galuh : 5
Ika : 10
Demi : 13
Chandra : 18
Janto : 23
Khailid : 29
Fandi : 31
Budi : 51
Hasan : 64
[2, 5, 6, 8, 10, 12, 32, 45]
[8, 14, 20, 26, 40]
[2, 5, 6, 8, 10, 12, 14, 20, 26, 32, 40, 45]
Bubble : 23.7104 detik
selection : 11.0129 detik
insertion : 9.64819 detik
merge : 0.176281 detik
quick : 0.0924933 detik
[12, 13, 45]
sorted : [124, 123, 15, 12, 4]
merge : 0.164328 detik
quick : 0.0969451 detik
merge mod : -0.02494 detik
quick mod : -0.241241 detik
List 1 :
3
7
12
13
16
List 2 :
1
9
10
Merged List :
1
3
7
9
10
12
13
16
>>>

Ln: 32 Col: 0
Ln: 25 Col: 0

Activate Windows
Go to Settings to activate Windows.
```

```
110_Modul6.py - C:\Users\Ramadhanu\Documents\Semester 4\110_Modul6.py (3.8.1)
File Edit Format Run Options Window Help

    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

mer = k[:1]
qui = k[:1]
mer2 = k[:1]
qui2 = k[:1]

aw=detak();mergeSort(mer);ak=detak();print('merge : %g detik' % (ak-aw));
aw=detak();quickSort(qui,0,len(qui)-1);ak=detak();print('quick : %g detik' % (ak-aw));
aw=detak();mergeSort(mer2);print('merge mod : %g detik' % (ak-aw));
aw=detak();quickSortMOD(qui2, False);print('quick mod : %g detik' % (ak-aw));

#Nomor 8
class Node:
    def __init__(self, data):
        self.data = data
        self.next = None

class LinkedList:
    def __init__(self):
        self.head = None
    def appendList(self, data):
        node = Node(data)
        if self.head == None:
            self.head = node
        else:
            curr = self.head
            while curr.next != None:
                curr = curr.next
            curr.next = node
    def appendSorted(self, data):
        node = Node(data)
        curr = self.head
        prev = None
        while curr is not None and curr.data < data:
            prev = curr
            curr = curr.next
        if prev == None:
            self.head = node
        else:
            prev.next = node

prev.next = node
node.next = curr

def printList(self):
    curr = self.head
    while curr != None:
        print ("%d"%curr.data),
        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 3.8.1 Shell
File Edit Shell Debug Options Window Help

>>>
===== RESTART: C:\Users\Ramadhanu\Documents\Semester 4\110_Modul6.py =====
Ahmad : 2
Eka : 4
Galuh : 5
Ika : 10
Demi : 13
Chandra : 18
Janto : 23
Khalid : 29
Fandi : 31
Budi : 51
Hasan : 64
[2, 5, 6, 10, 12, 32, 45]
[8, 14, 20, 26, 40]
[2, 5, 6, 8, 10, 12, 14, 20, 26, 32, 40, 45]
bubble : 23.7104 detik
selection : 11.0129 detik
insertion : 9.64819 detik
merge : 0.176281 detik
quick : 0.0924933 detik
[12, 13, 45]
sorted : [124, 123, 15, 12, 4]
merge : 0.164328 detik
quick : 0.0969651 detik
merge mod : -0.02494 detik
quick mod : -0.241241 detik
List 1 :
3
7
12
13
16
List 2 :
1
9
10
Merged List :
1
3
7
9
10
12
13
16
>>>
```

```
110_Modul6.py - C:\Users\Ramadhanu\Documents\Semester 4\110_Modul6.py (3.8.1)
File Edit Format Run Options Window Help

    prev.next = node
    node.next = curr

def printList(self):
    curr = self.head
    while curr != None:
        print ("%d"%curr.data),
        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 3.8.1 Shell
File Edit Shell Debug Options Window Help

>>>
===== RESTART: C:\Users\Ramadhanu\Documents\Semester 4\110_Modul6.py =====
Ahmad : 2
Eka : 4
Galuh : 5
Ika : 10
Demi : 13
Chandra : 18
Janto : 23
Khalid : 29
Fandi : 31
Budi : 51
Hasan : 64
[2, 5, 6, 10, 12, 32, 45]
[8, 14, 20, 26, 40]
[2, 5, 6, 8, 10, 12, 14, 20, 26, 32, 40, 45]
bubble : 23.7104 detik
selection : 11.0129 detik
insertion : 9.64819 detik
merge : 0.176281 detik
quick : 0.0924933 detik
[12, 13, 45]
sorted : [124, 123, 15, 12, 4]
merge : 0.164328 detik
quick : 0.0969651 detik
merge mod : -0.02494 detik
quick mod : -0.241241 detik
List 1 :
3
7
12
13
16
List 2 :
1
9
10
Merged List :
1
3
7
9
10
12
13
16
>>>
```