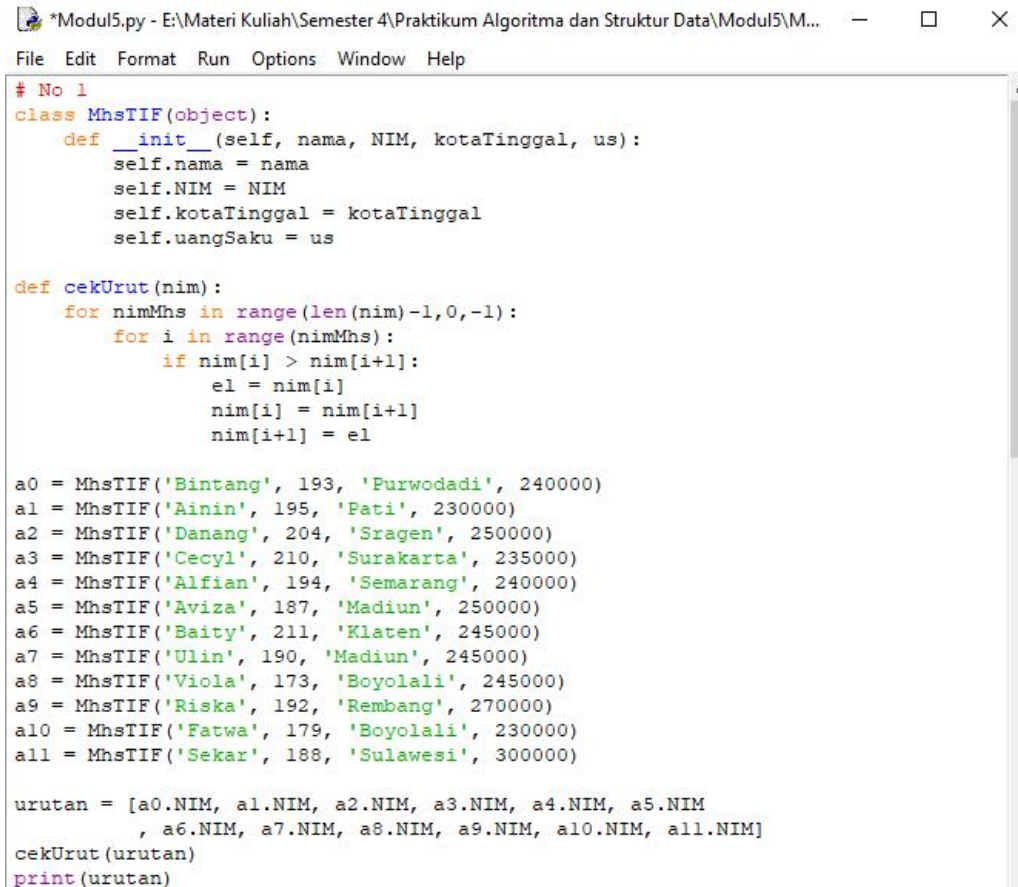


Nama : Amartya Bintang Wijat Ranti
NIM : L200180193
Kelas : G
Mata Kuliah : Praktikum Algoritma Struktur Data

Modul 5 : Pengurutan

Soal-Soal Untuk Mahasiswa

1. Buatlah suatu program untuk mengurutkan array mahasiswa berdasarkan NIM, yang elemennya tersebut dari class MhsTIF yang telah kamu buat sebelumnya.



```
# No 1
class MhsTIF(object):
    def __init__(self, nama, NIM, kotaTinggal, us):
        self.nama = nama
        self.NIM = NIM
        self.kotaTinggal = kotaTinggal
        self.uangSaku = us

    def cekUrut(nim):
        for nimMhs in range(len(nim)-1,0,-1):
            for i in range(nimMhs):
                if nim[i] > nim[i+1]:
                    el = nim[i]
                    nim[i] = nim[i+1]
                    nim[i+1] = el

a0 = MhsTIF('Bintang', 193, 'Purwodadi', 240000)
a1 = MhsTIF('Ainin', 195, 'Pati', 230000)
a2 = MhsTIF('Danang', 204, 'Sragen', 250000)
a3 = MhsTIF('Cecyl', 210, 'Surakarta', 235000)
a4 = MhsTIF('Alfian', 194, 'Semarang', 240000)
a5 = MhsTIF('Aviza', 187, 'Madiun', 250000)
a6 = MhsTIF('Baity', 211, 'Klaten', 245000)
a7 = MhsTIF('Ulin', 190, 'Madiun', 245000)
a8 = MhsTIF('Viola', 173, 'Boyolali', 245000)
a9 = MhsTIF('Riska', 192, 'Rembang', 270000)
a10 = MhsTIF('Fatwa', 179, 'Boyolali', 230000)
a11 = MhsTIF('Sekar', 188, 'Sulawesi', 300000)

urutan = [a0.NIM, a1.NIM, a2.NIM, a3.NIM, a4.NIM, a5.NIM,
           a6.NIM, a7.NIM, a8.NIM, a9.NIM, a10.NIM, a11.NIM]
cekUrut(urutan)
print(urutan)
```

Hasil Run :

```
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit (AMD64)]
on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: E:\Materi Kuliah\Semester 4\Praktikum Algoritma dan Struktur Data\Modul5\
Soal.py
[173, 179, 187, 188, 190, 192, 193, 194, 195, 204, 210, 211]
>>> |
```

- Misal terdapat 2 buah array yang sudah urut A dan B. buatlah suatu program untuk menggabungkan secara efisien kedua array itu menjadi suatu array C yang urut.

```
# No 2
def Kombinasi(angka):
    for number in range(len(angka)-1,0,-1):
        for i in range(number):
            if angka[i] > angka[i+1]:
                el = angka[i]
                angka[i] = angka[i+1]
                angka[i+1] = el

list1 = [0,2,4,6,8,12,13,15]
Kombinasi(list1)
a = list1

list2 = [1,3,5,7,9,10,11,14]
Kombinasi(list2)
b = list2

list3 = (a+b)
Kombinasi(list3)
c = list3
print(c)
```

Hasil Run :

```
>>>
RESTART: E:\Materi Kuliah\Semester 4\Praktikum Algoritma dan Struktur Data\Modul5\
Soal.py
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
>>> |
```

- Kamu mungkin dapat menduga, bubble sort lebih lambat dari selection sort dan juga insertion sort. Tapi manakah yang lebih cepat antara selection sort dan insertion sort? Untuk memulai menyelidikinya, kamu bisa membandingkan waktu yang diperlukan untuk mengurutkan sebuah array yang besar, minimal sepanjang 6000 (enam ribu) elemen.

```
time import time as detak
random import shuffle as kocok
ange (1,6001)
(k)
= k[:]
= k[:]
= k[:]

detak();bubbleSort(u_bub);ak = detak();print('bubble: %g detik' %(ak-aw));
detak();selectionSort(u_sel);ak = detak();print('selection: %g detik' %(ak-aw));
detak();insertionSort(u_ins);ak = detak();print('insertion: %g detik' %(ak-aw));|
```

```

# No 3
from time import time as detik
from random import shuffle as kocok

def bubbleSort(A):
    n = len(A)
    for i in range(n-1):
        for j in range(n-i-1):
            if A[j] > A[j+1]:
                swap(A,j,j+1)

def selectionSort(A):
    n = len(A)
    for i in range(n-1):
        indexKecil = cariPosisiYangTerkecil(A, i, n)
        if indexKecil != i:
            swap(A, i, indexKecil)

def insertionSort(A):
    n = len(A)
    for i in range(1,n):
        nilai = A[i]
        abc = i
        while abc > 0 and nilai < A[abc - 1]:
            A[abc] = A[abc - 1]
            abc = abc - 1
        A[abc] = nilai

def swap(A, a, b):
    tmp = A[a]
    A[a] = A[b]
    A[b] = tmp

def cariPosisiYangTerkecil(A, dariSini, sampaiSini):
    posisiYangTerkecil = dariSini
    for i in range(dariSini+1, sampaiSini):
        if A[i] < A[posisiYangTerkecil]:
            posisiYangTerkecil = i
    return posisiYangTerkecil

k = []
for i in range(1,6001):
    k.append(i)
kocok(k)

u_bub = k[:]
u_sel = k[:]
u_ins = k[:]

aw = detik();bubbleSort(u_bub);ak = detik();print('Bubble    |: %g detik' %(ak-aw));
aw = detik();selectionSort(u_sel);ak = detik();print('Selection : %g detik' %(ak-aw));
aw = detik();insertionSort(u_ins);ak = detik();print('Insertion : %g detik' %(ak-aw));

```

Hasil Run

```

>>>
RESTART: E:\Materi Kuliah\Semester 4\Praktikum Algoritma dan Struktur Data\Modul5\
Soal.py
Bubble    : 9.58362 detik
Selection : 2.68544 detik
Insertion : 4.03692 detik
>>> |

```