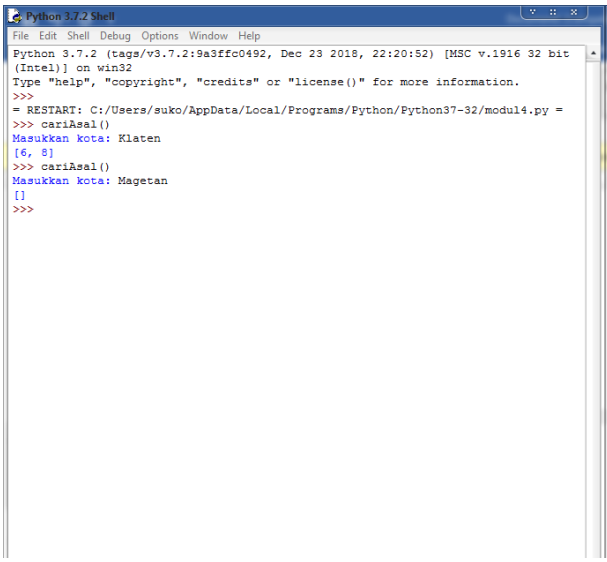
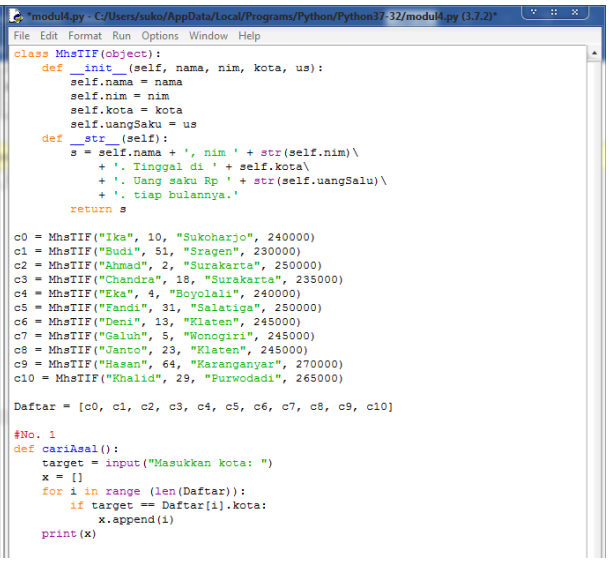
NAMA : PUJI NUGROHO
NIM : L200170123
KELAS : D
MODUL : 4

1.

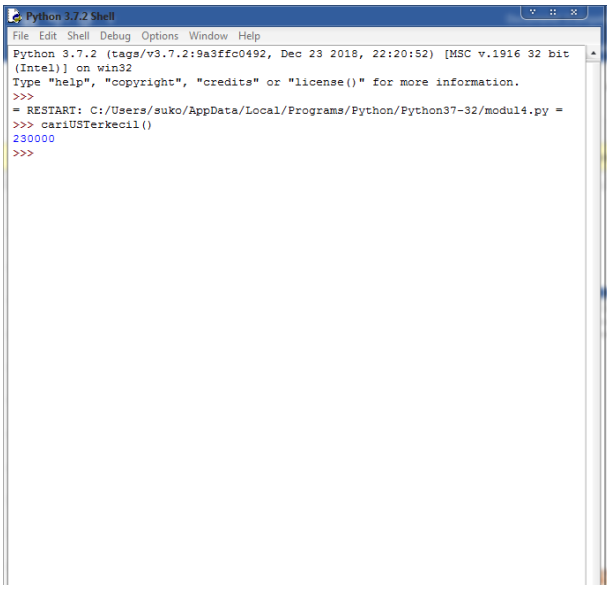


```
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:20:52) [MSC v.1916 32 bit
(Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/suko/AppData/Local/Programs/Python/Python37-32/modul4.py =
>>> cariAsal()
Masukkan kota: Klaten
[6, 8]
>>> cariAsal()
Masukkan kota: Magetan
[]
>>>
```

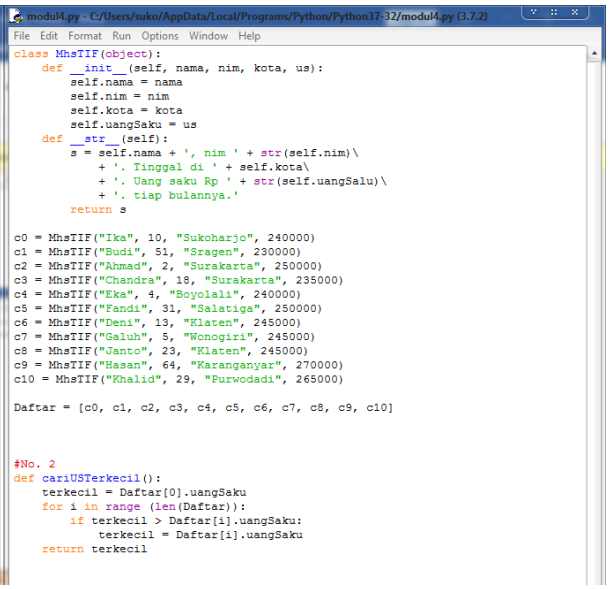


```
class MhsTIF(object):
    def __init__(self, nama, nim, kota, us):
        self.nama = nama
        self.nim = nim
        self.kota = kota
        self.uangSaku = us
    def __str__(self):
        s = self.nama + ', nim ' + str(self.nim)\
            + '. Tinggal di ' + self.kota\
            + '. Uang saku Rp ' + str(self.uangSalu)\
            + '. tiap bulannya.'
        return s

c0 = MhsTIF("Ika", 10, "Sukoharjo", 240000)
c1 = MhsTIF("Budi", 51, "Sragen", 230000)
c2 = MhsTIF("Ahmad", 2, "Surakarta", 250000)
c3 = MhsTIF("Chandra", 18, "Surakarta", 235000)
c4 = MhsTIF("Eka", 4, "Boyolali", 240000)
c5 = MhsTIF("Fandi", 31, "Salatiga", 250000)
c6 = MhsTIF("Deni", 13, "Klaten", 245000)
c7 = MhsTIF("Galuh", 5, "Wonogiri", 245000)
c8 = MhsTIF("Janto", 23, "Klaten", 245000)
c9 = MhsTIF("Hasan", 64, "Karanganyar", 270000)
c10 = MhsTIF("Khalid", 29, "Purwodadi", 265000)

Daftar = [c0, c1, c2, c3, c4, c5, c6, c7, c8, c9, c10]

#No. 1
def cariAsal():
    target = input("Masukkan kota: ")
    x = []
    for i in range (len(Daftar)):
        if target == Daftar[i].kota:
            x.append(i)
    print(x)
```

2.

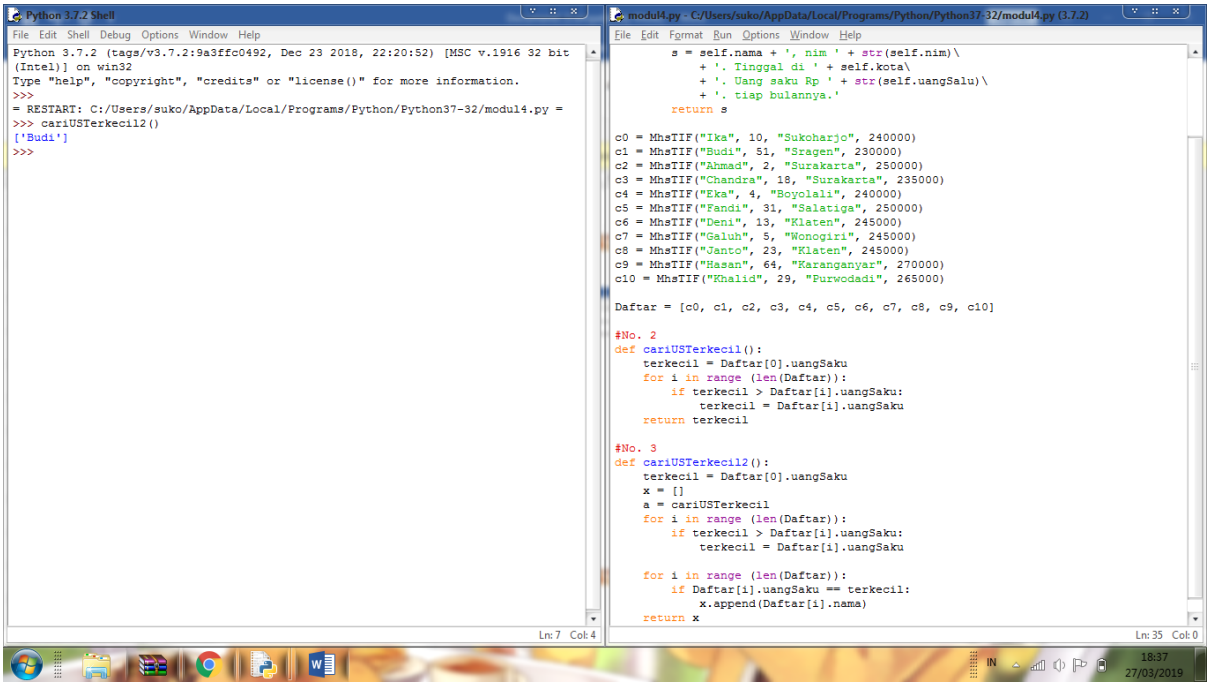


```
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:20:52) [MSC v.1916 32 bit
(Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/suko/AppData/Local/Programs/Python/Python37-32/modul4.py =
>>> cariUSTerkecil()
230000
>>>
```



```
class MhsTIF(object):
    def __init__(self, nama, nim, kota, us):
        self.nama = nama
        self.nim = nim
        self.kota = kota
        self.uangSaku = us
    def __str__(self):
        s = self.nama + ', nim ' + str(self.nim)\
            + '. Tinggal di ' + self.kota\
            + '. Uang saku Rp ' + str(self.uangSalu)\
            + '. tiap bulannya.'
        return s

c0 = MhsTIF("Ika", 10, "Sukoharjo", 240000)
c1 = MhsTIF("Budi", 51, "Sragen", 230000)
c2 = MhsTIF("Ahmad", 2, "Surakarta", 250000)
c3 = MhsTIF("Chandra", 18, "Surakarta", 235000)
c4 = MhsTIF("Eka", 4, "Boyolali", 240000)
c5 = MhsTIF("Fandi", 31, "Salatiga", 250000)
c6 = MhsTIF("Deni", 13, "Klaten", 245000)
c7 = MhsTIF("Galuh", 5, "Wonogiri", 245000)
c8 = MhsTIF("Janto", 23, "Klaten", 245000)
c9 = MhsTIF("Hasan", 64, "Karanganyar", 270000)
c10 = MhsTIF("Khalid", 29, "Purwodadi", 265000)

Daftar = [c0, c1, c2, c3, c4, c5, c6, c7, c8, c9, c10]


#No. 2
def cariUSTerkecil():
    terkecil = Daftar[0].uangSaku
    for i in range (len(Daftar)):
        if terkecil > Daftar[i].uangSaku:
            terkecil = Daftar[i].uangSaku
    return terkecil
```

3.



Python 3.7.2 Shell

```
File  Edit  Shell  Debug  Options  Window  Help
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: C:/Users/suko/AppData/Local/Programs/Python/Python37-32/modul4.py =
>>> cariUSTerkecil2()
['Budi']
>>>
```

Ln: 7 Col: 4

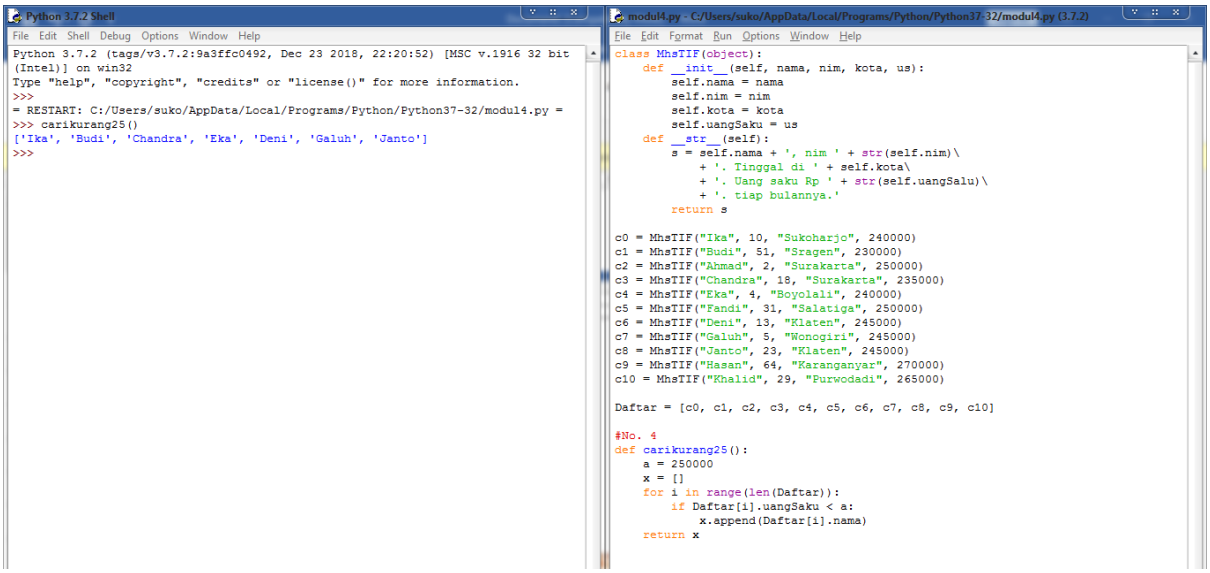
modul4.py - C:/Users/suko/AppData/Local/Programs/Python/Python37-32/modul4.py (3.7.2)

```
File  Edit  Format  Run  Options  Window  Help
        s = self.nama + ', nim ' + str(self.nim)\
            + '. Tinggal di ' + self.kota\
            + '. Uang saku Rp ' + str(self.uangSalu)\
            + '. tiap bulannya.'
        return s

c0 = MhsTIF("Ika", 10, "Sukoharjo", 240000)
c1 = MhsTIF("Budi", 51, "Sragen", 230000)
c2 = MhsTIF("Ahmad", 2, "Surakarta", 250000)
c3 = MhsTIF("Chandra", 18, "Surakarta", 235000)
c4 = MhsTIF("Eka", 4, "Boyolali", 240000)
c5 = MhsTIF("Fandi", 31, "Salatiga", 250000)
c6 = MhsTIF("Deni", 13, "Klaten", 245000)
c7 = MhsTIF("Galuh", 5, "Wonogiri", 245000)
c8 = MhsTIF("Janto", 23, "Klaten", 245000)
c9 = MhsTIF("Hasan", 64, "Karanganyar", 270000)
c10 = MhsTIF("Khalid", 29, "Purwodadi", 265000)

Daftar = [c0, c1, c2, c3, c4, c5, c6, c7, c8, c9, c10]

#No. 2
def cariUSTerkecil():
    terkecil = Daftar[0].uangSaku
    for i in range (len(Daftar)):
        if terkecil > Daftar[i].uangSaku:
            terkecil = Daftar[i].uangSaku
    return terkecil

#No. 3
def cariUSTerkecil2():
    terkecil = Daftar[0].uangSaku
    x = []
    a = cariUSTerkecil
    for i in range (len(Daftar)):
        if terkecil > Daftar[i].uangSaku:
            terkecil = Daftar[i].uangSaku

    for i in range (len(Daftar)):
        if Daftar[i].uangSaku == terkecil:
            x.append(Daftar[i].nama)
    return x
```

Ln: 35 Col: 0

IN 18:37 27/03/2019

4.



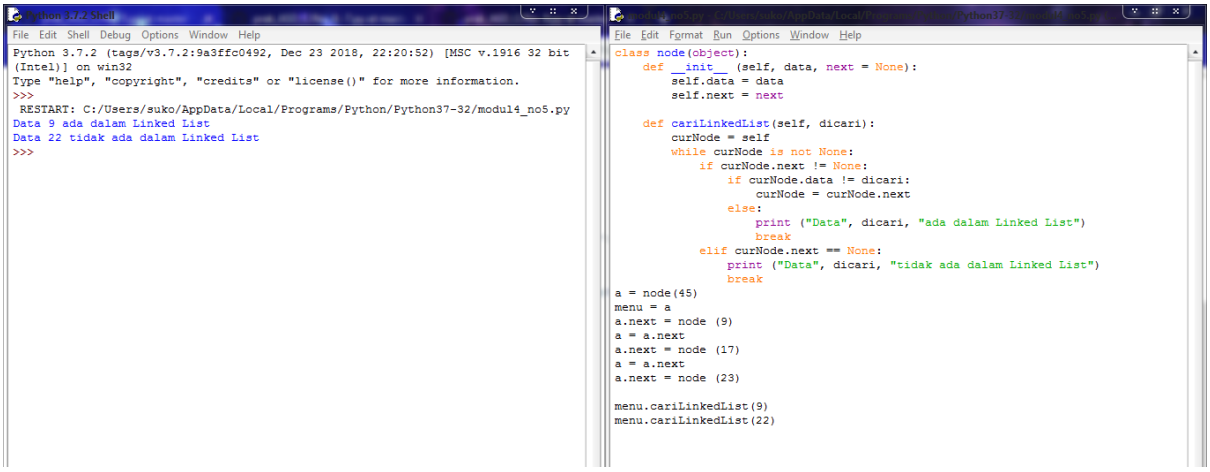
Python 3.7.2 Shell

```
File  Edit  Shell  Debug  Options  Window  Help
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: C:/Users/suko/AppData/Local/Programs/Python/Python37-32/modul4.py =
>>> carikurang25()
['Ika', 'Budi', 'Chandra', 'Eka', 'Deni', 'Galuh', 'Janto']
>>>
```

modul4.py - C:/Users/suko/AppData/Local/Programs/Python/Python37-32/modul4.py (3.7.2)

```
File  Edit  Format  Run  Options  Window  Help
class MhsTIF(object):
    def __init__(self, nama, nim, kota, us):
        self.nama = nama
        self.nim = nim
        self.kota = kota
        self.uangSaku = us
    def __str__(self):
        s = self.nama + ', nim ' + str(self.nim)\
            + '. Tinggal di ' + self.kota\
            + '. Uang saku Rp ' + str(self.uangSalu)\
            + '. tiap bulannya.'
        return s

c0 = MhsTIF("Ika", 10, "Sukoharjo", 240000)
c1 = MhsTIF("Budi", 51, "Sragen", 230000)
c2 = MhsTIF("Ahmad", 2, "Surakarta", 250000)
c3 = MhsTIF("Chandra", 18, "Surakarta", 235000)
c4 = MhsTIF("Eka", 4, "Boyolali", 240000)
c5 = MhsTIF("Fandi", 31, "Salatiga", 250000)
c6 = MhsTIF("Deni", 13, "Klaten", 245000)
c7 = MhsTIF("Galuh", 5, "Wonogiri", 245000)
c8 = MhsTIF("Janto", 23, "Klaten", 245000)
c9 = MhsTIF("Hasan", 64, "Karanganyar", 270000)
c10 = MhsTIF("Khalid", 29, "Purwodadi", 265000)

Daftar = [c0, c1, c2, c3, c4, c5, c6, c7, c8, c9, c10]

#No. 4
def carikurang25():
    a = 250000
    x = []
    for i in range(len(Daftar)):
        if Daftar[i].uangSaku < a:
            x.append(Daftar[i].nama)
    return x
```

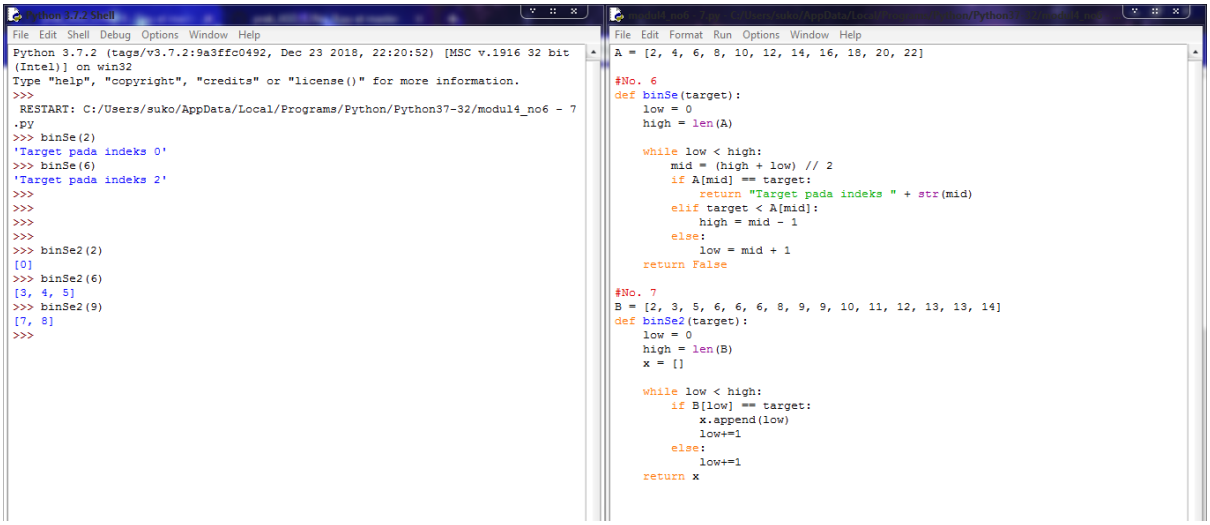
5.



```
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: C:/Users/suko/AppData/Local/Programs/Python/Python37-32/modul4_no5.py
Data 9 ada dalam Linked List
Data 22 tidak ada dalam Linked List
>>>
```

```
class node(object):
    def __init__ (self, data, next = None):
        self.data = data
        self.next = next

    def cariLinkedList(self, dicari):
        curNode = self
        while curNode is not None:
            if curNode.next != None:
                if curNode.data != dicari:
                    curNode = curNode.next
                else:
                    print ("Data", dicari, "ada dalam Linked List")
                    break
            elif curNode.next == None:
                print ("Data", dicari, "tidak ada dalam Linked List")
                break
a = node(45)
menu = a
a.next = node (9)
a = a.next
a.next = node (17)
a = a.next
a.next = node (23)

menu.cariLinkedList(9)
menu.cariLinkedList(22)
```

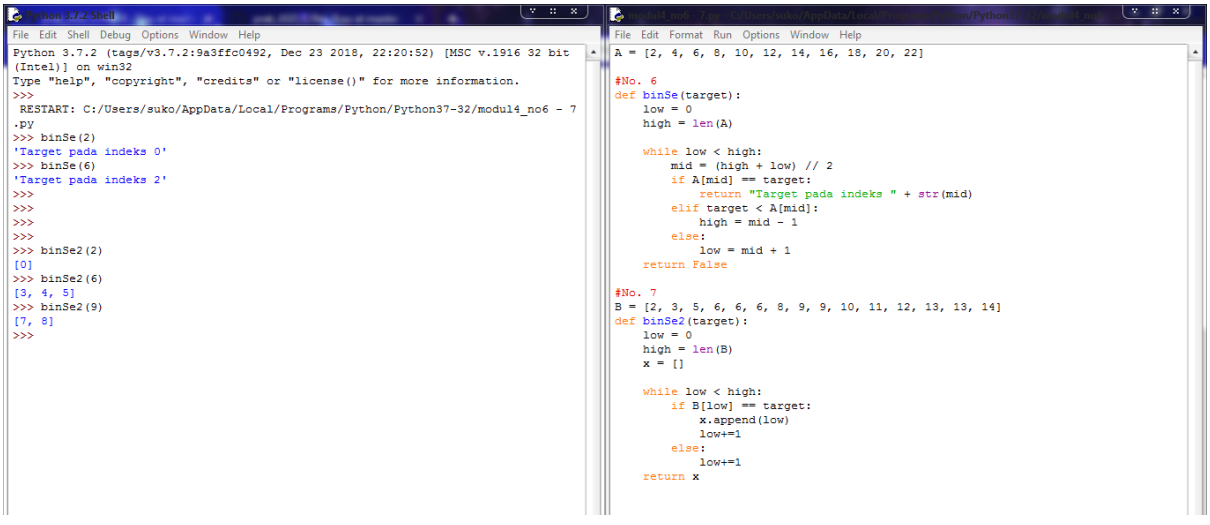
6.



```
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: C:/Users/suko/AppData/Local/Programs/Python/Python37-32/modul4_no6 - 7
.py
>>> binSe(2)
'Target pada indeks 0'
>>> binSe(6)
'Target pada indeks 2'
>>>
>>>
>>>
>>>
>>> binSe2(2)
[0]
>>> binSe2(6)
[3, 4, 5]
>>> binSe2(9)
[7, 8]
>>>
```

```
A = [2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22]

#No. 6
def binSe(target):
    low = 0
    high = len(A)

    while low < high:
        mid = (high + low) // 2
        if A[mid] == target:
            return "Target pada indeks " + str(mid)
        elif target < A[mid]:
            high = mid - 1
        else:
            low = mid + 1
    return False

#No. 7
B = [2, 3, 5, 6, 6, 6, 8, 9, 9, 10, 11, 12, 13, 13, 14]
def binSe2(target):
    low = 0
    high = len(B)
    x = []

    while low < high:
        if B[low] == target:
            x.append(low)
            low+=1
        else:
            low+=1
    return x
```

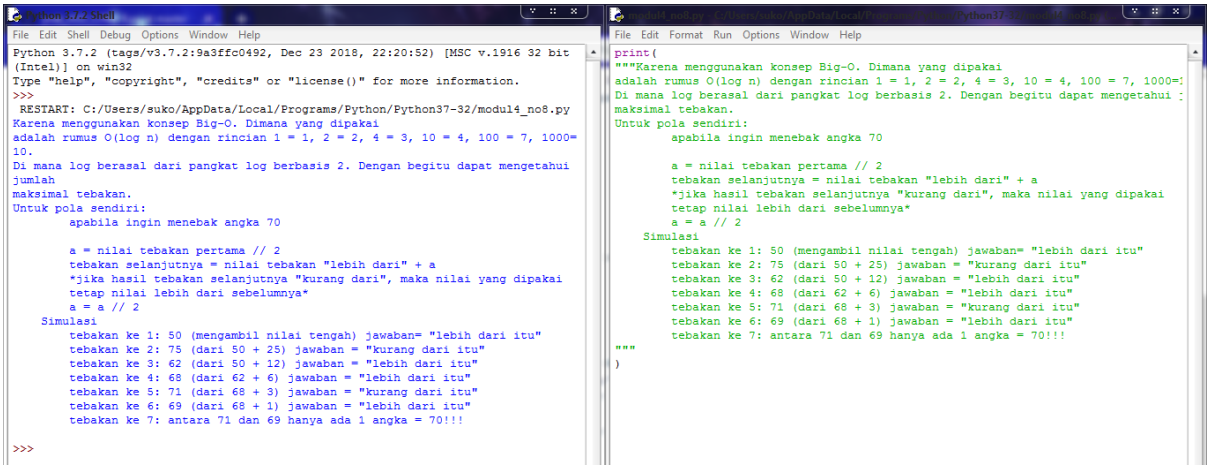
7.



```
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: C:/Users/suko/AppData/Local/Programs/Python/Python37-32/modul4_no6 - 7
.py
>>> binSe(2)
'Target pada indeks 0'
>>> binSe(6)
'Target pada indeks 2'
>>>
>>>
>>>
>>>
>>> binSe2(2)
[0]
>>> binSe2(6)
[3, 4, 5]
>>> binSe2(9)
[7, 8]
>>>
```

```
A = [2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22]

#No. 6
def binSe(target):
    low = 0
    high = len(A)

    while low < high:
        mid = (high + low) // 2
        if A[mid] == target:
            return "Target pada indeks " + str(mid)
        elif target < A[mid]:
            high = mid - 1
        else:
            low = mid + 1
    return False

#No. 7
B = [2, 3, 5, 6, 6, 6, 8, 9, 9, 10, 11, 12, 13, 13, 14]
def binSe2(target):
    low = 0
    high = len(B)
    x = []

    while low < high:
        if B[low] == target:
            x.append(low)
            low+=1
        else:
            low+=1
    return x
```

8.



```
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: C:/Users/suko/AppData/Local/Programs/Python/Python37-32/modul4_no8.py
Karena menggunakan konsep Big-O. Dimana yang dipakai
adalah rumus O(log n) dengan rincian 1 = 1, 2 = 2, 4 = 3, 10 = 4, 100 = 7, 1000=
10.
Di mana log berasal dari pangkat log berbasis 2. Dengan begitu dapat mengetahui
jumlah
maksimal tebakan.
Untuk pola sendiri:
        apabila ingin menebak angka 70

        a = nilai tebakan pertama // 2
        tebakan selanjutnya = nilai tebakan "lebih dari" + a
        *jika hasil tebakan selanjutnya "kurang dari", maka nilai yang dipakai
        tetap nilai lebih dari sebelumnya*
        a = a // 2
    Simulasi
        tebakan ke 1: 50 (mengambil nilai tengah) jawaban= "lebih dari itu"
        tebakan ke 2: 75 (dari 50 + 25) jawaban = "kurang dari itu"
        tebakan ke 3: 62 (dari 50 + 12) jawaban = "lebih dari itu"
        tebakan ke 4: 68 (dari 62 + 6) jawaban = "lebih dari itu"
        tebakan ke 5: 71 (dari 68 + 3) jawaban = "kurang dari itu"
        tebakan ke 6: 69 (dari 68 + 1) jawaban = "lebih dari itu"
        tebakan ke 7: antara 71 dan 69 hanya ada 1 angka = 70!!!

>>>
```

```
print(
"""Karena menggunakan konsep Big-O. Dimana yang dipakai
adalah rumus O(log n) dengan rincian 1 = 1, 2 = 2, 4 = 3, 10 = 4, 100 = 7, 1000=1
Di mana log berasal dari pangkat log berbasis 2. Dengan begitu dapat mengetahui j
maksimal tebakan.
Untuk pola sendiri:
        apabila ingin menebak angka 70

        a = nilai tebakan pertama // 2
        tebakan selanjutnya = nilai tebakan "lebih dari" + a
        *jika hasil tebakan selanjutnya "kurang dari", maka nilai yang dipakai
        tetap nilai lebih dari sebelumnya*
        a = a // 2
    Simulasi
        tebakan ke 1: 50 (mengambil nilai tengah) jawaban= "lebih dari itu"
        tebakan ke 2: 75 (dari 50 + 25) jawaban = "kurang dari itu"
        tebakan ke 3: 62 (dari 50 + 12) jawaban = "lebih dari itu"
        tebakan ke 4: 68 (dari 62 + 6) jawaban = "lebih dari itu"
        tebakan ke 5: 71 (dari 68 + 3) jawaban = "kurang dari itu"
        tebakan ke 6: 69 (dari 68 + 1) jawaban = "lebih dari itu"
        tebakan ke 7: antara 71 dan 69 hanya ada 1 angka = 70!!!
"""
)
```

Penjelasan dalam bentuk teks :

Karena menggunakan konsep Big-O. Dimana yang dipakai
adalah rumus O(log n) dengan rincian 1 = 1, 2 = 2, 4 = 3, 10 = 4, 100 = 7, 1000=10.
Di mana log berasal dari pangkat log berbasis 2. Dengan begitu dapat mengetahui jumlah maksimal tebakan.

Untuk pola sendiri:

apabila ingin menebak angka 70

a = nilai tebakan pertama // 2
tebakan selanjutnya = nilai tebakan "lebih dari" + a

*jika hasil tebakan selanjutnya "kurang dari", maka nilai yang dipakai
tetap nilai lebih dari sebelumnya*

a = a // 2

Simulasi

tebakan ke 1: 50 (mengambil nilai tengah) jawaban= "lebih dari itu"
tebakan ke 2: 75 (dari 50 + 25) jawaban = "kurang dari itu"
tebakan ke 3: 62 (dari 50 + 12) jawaban = "lebih dari itu"
tebakan ke 4: 68 (dari 62 + 6) jawaban = "lebih dari itu"
tebakan ke 5: 71 (dari 68 + 3) jawaban = "kurang dari itu"
tebakan ke 6: 69 (dari 68 + 1) jawaban = "lebih dari itu"
tebakan ke 7: antara 71 dan 69 hanya ada 1 angka = 70!!!