Nama : Veny Fitriana Isnaini
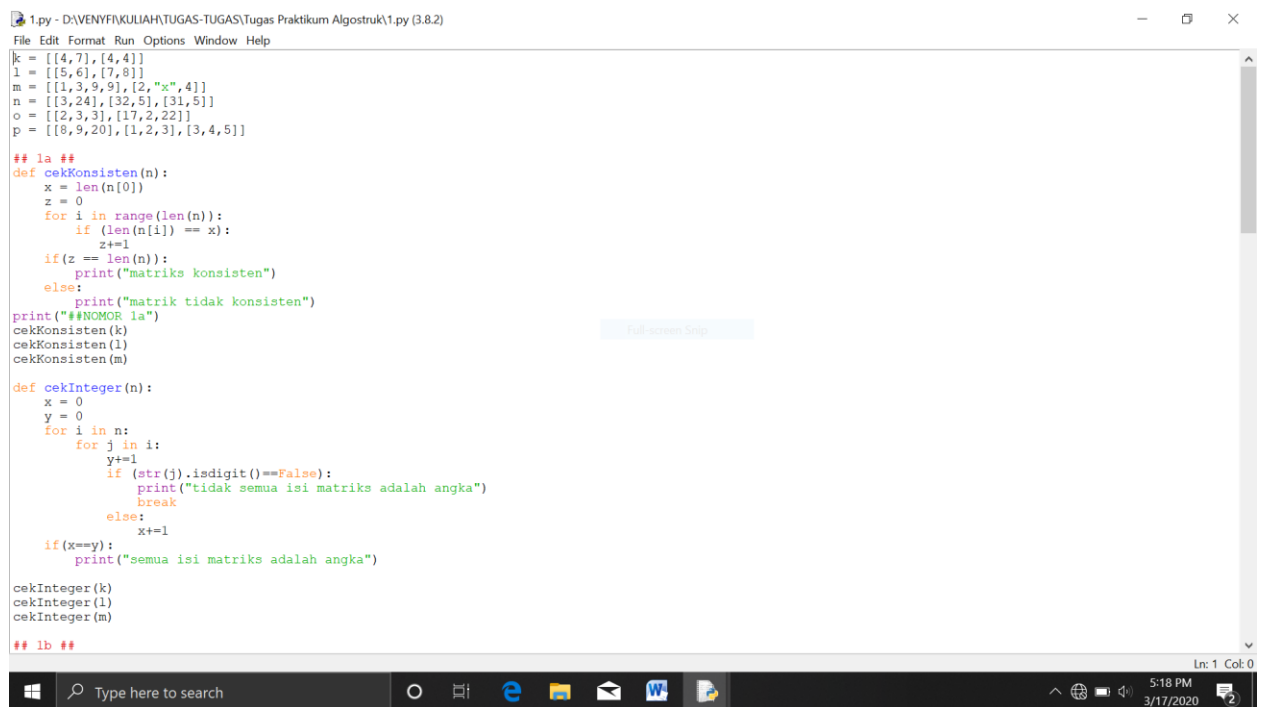
NIM : L200180045

Kelas : B

Modul 3

Collections, Arrays, and Linked Structures

1.


```
k = [[4,7],[4,4]]
l = [[5,6],[7,8]]
m = [[1,3,9,9],[2,"x",4]]
n = [[3,24],[32,5],[31,5]]
o = [[2,3,3],[17,2,22]]
p = [[8,9,20],[1,2,3],[3,4,5]]

## la ##
def cekKonsisten(n):
    x = len(n[0])
    z = 0
    for i in range(len(n)):
        if (len(n[i]) == x):
            z+=1
    if(z == len(n)):
        print("matriks konsisten")
    else:
        print("matrik tidak konsisten")
print("##NOMOR la")
cekKonsisten(k)
cekKonsisten(l)
cekKonsisten(m)

def cekInteger(n):
    x = 0
    y = 0
    for i in n:
        for j in i:
            y+=1
            if (str(j).isdigit()==False):
                print("tidak semua isi matriks adalah angka")
                break
            else:
                x+=1
    if(x==y):
        print("semua isi matriks adalah angka")

cekInteger(k)
cekInteger(l)
cekInteger(m)

## lb ##
```

```python
## 1b ##
def cekordo(n):
    x,y = 0,0
    for i in range(len(n)):
        x+=1
        y = len(n[i])
    print("mempunyai ordo "+str(x)+"x"+str(y))

print("##NOMOR 1b")
cekordo(k)
cekordo(l)
cekordo(m)
cekordo(n)

## 1c ##
def jumlah(n,m):
    x,y = 0,0
    for i in range(len(n)):
        x+=1
        y = len(n[i])
    xy = [[0 for j in range(x)] for i in range(y)]

    z = 0
    if(len(n)==len(m)):
        for i in range(len(n)):
            if(len(n[i]) == len(m[i])):
                z+=1
    if(z==len(n) and z==len(m)):
        print("ukuran sama")
        for i in range(len(n)):
            for j in range(len(n[i])):
                xy[i][j] = n[i][j] + m[i][j]
        print(xy)
    else:
        print("ukuran beda")

print("##NOMOR 1c")
jumlah(k,l)
jumlah(k,n)

## 1d ##
```

```python
## 1d ##
def kali(n,m):
    aa = 0
    x,y = 0,0
    for i in range(len(n)):
        x+=1
        y = len(n[i])
    v,w = 0,0
    for i in range(len(m)):
        v+=1
        w = len(m[i])

    if(y==v):
        print("bisa dikalikan")
        vwxy = [[0 for j in range(w)] for i in range(x)]
        for i in range(len(n)):
            for j in range(len(m[0])):
                for k in range(len(m)):
                    #print(n[i][k], m[k][j])
                    vwxy[i][j] += n[i][k] * m[k][j]
        print(vwxy)

    else:
        print("tidak memenuhi syarat")
print("##NOMOR 1d")
zz = [[1,2,3],[1,2,3]]
zx = [[1],[2],[3]]
kali(zz,zx)
kali(k,l)
kali(k,o)
kali(k,zx)

## 1e ##
def determinanHitung(A, total=0):
    x = len(A[0])
    z = 0
    for i in range(len(A)):
        if (len(A[i]) == x):
            z+=1
    if(z == len(A)):
        if(x==len(A)):
```

File Edit Format Run Options Window Help

```python
## le ##
def determinanHitung(A, total=0):
    x = len(A[0])
    z = 0
    for i in range(len(A)):
        if (len(A[i]) == x):
            z+=1
    if(z == len(A)):
        if(x==len(A)):
            indices = list(range(len(A)))
            if len(A) == 2 and len(A[0]) == 2:
                val = A[0][0] * A[1][1] - A[1][0] * A[0][1]
                return val
            for fc in indices:
                As = A
                As = As[1:]
                height = len(As)
                for i in range(height):
                    As[i] = As[i][0:fc] + As[i][fc+1:]
                sign = (-1) ** (fc % 2)
                sub_det = determinanHitung(As)
                total += sign * A[0][fc] * sub_det
        else:
            return "tidak bisa dihitung determinannya, karena bukan matrix bujursangkar"
    else:
        return "tidak bisa dihitung determinannya, karena bukan matrix bujursangkar"
    return total


q = [[3,1],[2,5]]
r = [[1,2,1],[3,3,1],[2,1,2]]
s = [[1,-2,0,0],[3,2,-3,1],[4,0,5,1],[2,3,-1,4]]
t = [[10,23,45,12,13],[1,2,3,4,5],[1,2,3,4,6],[4,2,3,4,8],[1,4,5,6,10]]

print("##NOMOR 1e")
print(determinanHitung(q))
print(determinanHitung(r))
print(determinanHitung(s))
print(determinanHitung(t))
print(determinanHitung(n))
print(determinanHitung(o))
```

Ln: 1 Col: 0

File Edit Shell Debug Options Window Help

```
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:\VENYFI\KULIAH\TUGAS-TUGAS\Tugas Praktikum Algostruk\1.py =====
##NOMOR 1a
matriks konsisten
matriks konsisten
matrik tidak konsisten
semua isi matriks adalah angka
semua isi matriks adalah angka
tidak semua isi matriks adalah angka
##NOMOR 1b
mempunyai ordo 2x2
mempunyai ordo 2x2
mempunyai ordo 2x3
mempunyai ordo 3x2
##NOMOR 1c
ukuran sama
[[9, 13], [11, 12]]
ukuran beda
##NOMOR 1d
bisa dikalikan
[[14], [14]]
bisa dikalikan
[[69, 80], [48, 56]]
bisa dikalikan
[[127, 26, 166], [76, 20, 100]]
tidak memenuhi syarat
##NOMOR 1e
13
-6
200
330
tidak bisa dihitung determinannya, karena bukan matrix bujursangkar
tidak bisa dihitung determinannya, karena bukan matrix bujursangkar
>>>
```

Ln: 36 Col: 4

2.


```python
## 2a ##
def buatNol(n,m=None):
    if(m==None):
        m=n
    print("Membuat matriks 0 dengan ordo "+str(n)+"x"+str(m))
    print([[0 for x in range(m)] for y in range(n)])
print("##NOMOR 2a")
buatNol(3)
buatNol(4,3)

## 2b ##
def buatIdentitas(n):
    print("Membuat matriks Identitas dengan ordo"+str(n)+"x"+str(n))
    print([[1 if j==i else 0 for j in range(n)] for i in range(n)])

print("##NOMOR 2b")
buatIdentitas(5)
buatIdentitas(3)
```



```
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:/VENYFI/KULIAH/TUGAS-TUGAS/Tugas Praktikum Algostruk/2.py =====
##NOMOR 2a
Membuat matriks 0 dengan ordo 3x3
[[0, 0, 0], [0, 0, 0], [0, 0, 0]]
Membuat matriks 0 dengan ordo 4x3
[[0, 0, 0], [0, 0, 0], [0, 0, 0], [0, 0, 0]]
##NOMOR 2b
Membuat matriks Identitas dengan ordo5x5
[[1, 0, 0, 0, 0], [0, 1, 0, 0, 0], [0, 0, 1, 0, 0], [0, 0, 0, 1, 0], [0, 0, 0, 0, 1]]
Membuat matriks Identitas dengan ordo3x3
[[1, 0, 0], [0, 1, 0], [0, 0, 1]]
>>>
```

3.

```python
class Node:
    def __init__(self, data):
        self.data = data
        self.next = None
class LinkedList:
    def __init__(self):
        self.head = None
    def pushAwal(self, new_data):
        new_node = Node(new_data)
        new_node.next = self.head
        self.head = new_node
    def pushAkhir(self, data):
        if (self.head == None):
            self.head = Node(data)
        else:
            current = self.head
            while (current.next != None):
                current = current.next
            current.next = Node(data)
        return self.head
    def tambah(self,data,pos):
        node = Node(data)
        if not self.head:
            self.head = node
        elif pos==0:
            node.next = self.head
            self.head = node
        else:
            prev = None
            current = self.head
            current_pos = 0
            while(current_pos < pos) and current.next:
                prev = current
                current = current.next
                current_pos +=1
            prev.next = node
            node.next = current
        return self.head
    def hapus(self, position):
        if self.head == None:
            return
        temp = self.head
```

Ln: 80 Col: 16

```python
        temp = self.head
        if position == 0:
            self.head = temp.next
            temp = None
            return
        for i in range(position -1 ):
            temp = temp.next
            if temp is None:
                break
        if temp is None:
            return
        if temp.next is None:
            return
        next = temp.next.next
        temp.next = None
        temp.next = next
    def cari(self, x):
        current = self.head
        while current != None:
            if current.data == x:
                return "True"
            current = current.next
        return "False"
    def display(self):
        current = self.head
        while current is not None:
            print(current.data, end = ' ')
            current = current.next

llist = LinkedList()
llist.pushAwal(12)
llist.pushAwal(13)
llist.pushAwal(14)
llist.pushAwal(15)
llist.pushAwal(3)
llist.pushAwal(17)
llist.pushAkhir(18)
llist.hapus(0)
llist.tambah(1,4)
print(llist.cari(13))
print(llist.cari(16))
llist.display()
```

Ln: 83 Col: 15

File Edit Shell Debug Options Window Help

```
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:/VENYFI/KULIAH/TUGAS-TUGAS/Tugas Praktikum Algostruk/3.py =====
True
False
3 15 14 13 1 12 18
>>>
```

4.

File Edit Format Run Options Window Help

```python
class Node:
    def __init__(self, data):
        self.data = data
        self.prev = None
class DoublyLinkedList:
    def __init__(self):
        self.head = None
    def menambahAwal(self, new_data):
        print("menambah pada awal", new_data)
        new_node = Node(new_data)
        new_node.next = self.head
        if self.head is not None:
            self.head.prev = new_node
        self.head = new_node
    def menambahAkhir(self, new_data):
        print("menambah pada akhir", new_data)
        new_node = Node(new_data)
        new_node.next = None
        if self.head is None:
            new_node.prev = None
            self.head = new_node
            return
        last = self.head
        while(last.next is not None):
            last = last.next
        last.next = new_node
        new_node.prev = last
        return
    def printList(self, node):
        print("\nDari Depan :")
        while(node is not None):
            print(" % d" %(node.data))
            last = node
            node = node.next
        print("\nDari Belakang :")
        while(last is not None):
            print(" % d" %(last.data))
            last = last.prev
llist = DoublyLinkedList()
llist.menambahAwal(2)
llist.menambahAwal(1)
llist.menambahAkhir(3)
```

```
Python 3.8.2 Shell                                                    —  □  ✕
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:/VENYFI/KULIAH/TUGAS-TUGAS/Tugas Praktikum Algostruk/4.py =====
menambah pada awal 2
menambah pada awal 1
menambah pada akhir 3
menambah pada akhir 4

Dari Depan :
  1
  2
  3
  4

Dari Belakang :
  4
  3
  2
  1
>>> |
```