Muhibah Fata Tika
L200170156
D
Praktikum ALGOSTRUK
Modul 3

TUGAS PRAKTIKUM
ALGORITMA DAN STRUKTUR DATA
MODUL3

1. Array 2 Dimensi

a. Mengecek apakah matriks tersebut konsisten dan cek tipe data

```
#a
def cek(v,n):
    a = [[v for i in range(v)] for j in range(n)]
    for i in a:
        print (i)
    if v == n:
        print('konsisten')
    else:
        print ('tidak konsisten')
```

```
>>> cek(3,3)
[3, 3, 3]
[3, 3, 3]
[3, 3, 3]
konsisten
>>> cek(3,2)
[3, 3, 3]
[3, 3, 3]
tidak konsisten
>>>
```

b. Mengambil ukuran matriks

```
#b
def ukuran(v,n):
    a = [[v for i in range(v)] for j in range(n)]
    for i in a:
        print (i)
    print('matriks ',v, ' x ' ,n)
```

Muhibah Fata Tika
L200170156
D
Praktikum ALGOSTRUK
Modul 3

```
>>> ukuran(2,2)
[2, 2]
[2, 2]
matriks  2  x  2
```

c. Menjumlahkan 2 matriks

```
#c
def menjumlahkan(v,n):
    a = [[n for i in range(v)] for j in range(n)]
    for i in a:
        print (i)
    print('\n',' + \n')

    b = [[v for i in range(v)] for j in range(n)]
    for i in b:
        print (i)
    print('\n','Hasil dari penjumlahan adalah ')

    c = v + n
    d = [[c for i in range(v)] for j in range(n)]
    for i in d:
        print (i)
```

```
>>> menjumlahkan(2,3)
[3, 3]
[3, 3]
[3, 3]

  +

[2, 2]
[2, 2]
[2, 2]

 Hasil dari penjumlahan adalah
[5, 5]
[5, 5]
[5, 5]
```

d. Mengalikan 2 matriks

Muhibah Fata Tika
L200170156
D
Praktikum ALGOSTRUK
Modul 3

```
#d
def mengalikan(v,n):
    a = [[n for i in range(v)] for j in range(n)]
    for i in a:
        print (i)
    print('\n',' x \n')

    b = [[v for i in range(v)] for j in range(n)]
    for i in b:
        print (i)
    print('\n','Hasil dari perkalian adalah ')

    c = v * n
    d = [[c for i in range(v)] for j in range(n)]
    for i in d:
        print (i)
```

```
>>> mengalikan(4,4)
[4, 4, 4, 4]
[4, 4, 4, 4]
[4, 4, 4, 4]
[4, 4, 4, 4]

  x

[4, 4, 4, 4]
[4, 4, 4, 4]
[4, 4, 4, 4]
[4, 4, 4, 4]

 Hasil dari perkalian adalah
[16, 16, 16, 16]
[16, 16, 16, 16]
[16, 16, 16, 16]
[16, 16, 16, 16]
```

e. Menghitung determinan

Muhibah Fata Tika
L200170156
D
Praktikum ALGOSTRUK
Modul 3

```
def determinan(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 = determinan(As)
                total += sign * A[0][fc] * sub_det
        else:
            return "tidak bisa dihitung determinan, bukan matrix bujursangkar"
    else:
        return "tidak bisa dihitung determinan, bukan matrix bujursangkar"
    return total


z = [[3,1],[2,5]]
x = [[1,2,1],[3,3,1],[2,1,2]]
v = [[1,-2,0,0],[3,2,-3,1],[4,0,5,1],[2,3,-1,4]]
r = [[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]]
d = [[3,4],[2,4],[1,5]]
e = [[5,6,7],[7,8,9]]
print("Ini hasil yang determinan")
print(determinan(z))
print(determinan(x))
print(determinan(v))
print(determinan(r))
print(determinan(d))
print(determinan(e))
```

```
Ini hasil yang determinan
13
-6
200
330
tidak bisa dihitung determinan, bukan matrix bujursangkar
tidak bisa dihitung determinan, bukan matrix bujursangkar
```

2. List Comprehension
   Membuat matriks 0 dan matriks identitas

Muhibah Fata Tika
L200170156
D
Praktikum ALGOSTRUK
Modul 3

```
def buatNol(n,m=None):
    if(m==None):
        m=n
    print("membuat matriks 0 dengan ordo "+str(n)+"x"+str(m))
    print([[0 for j in range(m)] for i in range(n)])

buatNol(2,4)
buatNol(3)

def buatIden(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)])

buatIden(4)
buatIden(2)
```

```
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========= RESTART: D:/Senester 4/Prak ALGOSTRUK/ModulKe3_D_156/nol.py =========
membuat matriks 0 dengan ordo 2x4
[[0, 0, 0, 0], [0, 0, 0, 0]]
membuat matriks 0 dengan ordo 3x3
[[0, 0, 0], [0, 0, 0], [0, 0, 0]]
membuat matriks identitas dengan ordo4x4
[[1, 0, 0, 0], [0, 1, 0, 0], [0, 0, 1, 0], [0, 0, 0, 1]]
membuat matriks identitas dengan ordo2x2
[[1, 0], [0, 1]]
>>>
```

3. Linked List
    - Mencari data tertentu
    - Menambah simpul di awal dan diakhir
    - Menyisipkan simpul di posisi tertentu
    - Menghapus simpul di posisi tertentu

Muhibah Fata Tika
L200170156
D
Praktikum ALGOSTRUK
Modul 3

```
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> class Node:
    def __init__(self, data):
        self.data = data
        self.next = None
class LinkedList:
    def __init__(self):
        self.head = None
    def pushAw(self, new_data):
        new_node = Node(new_data)
        new_node.next = self.head
        self.head = new_node
    def pushAk(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 insert(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
```

Muhibah Fata Tika
L200170156
D
Praktikum ALGOSTRUK
Modul 3

```
            prev.next = node
            node.next = current
        return self.head
    def deleteNode(self, position):
        if self.head == None:
            return
        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 search(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.pushAw(21)
llist.pushAw(22)
llist.pushAw(12)
llist.pushAw(14)
llist.pushAw(2)
llist.pushAw(19)
llist.pushAk(9)
llist.deleteNode(0)
llist.insert(1,6)
print(llist.search(21))
print(llist.search(29))
llist.display()
```

4. Doubly Linked List
    - Mengunjungi dan mencetak dari depan dan belakang
    - Mengambah simpul di awal dan di akhir

Muhibah Fata Tika
L200170156
D
Praktikum ALGOSTRUK
Modul 3

```
class DoublyLinkedList:
    def __init__(self):
        self.head = None
    def awal(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 akhir(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.awal(7)
llist.awal(1)
llist.akhir(6)
llist.akhir(4)
llist.printList(llist.head)
```

```
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
======== RESTART: D:/Senester 4/Prak ALGOSTRUK/ModulKe3_D_156/no4.py ========
menambah pada awal 7
menambah pada awal 1
menambah pada akhir 6
menambah pada akhir 4

Dari Depan :
  1
  7
  6
  4

Dari Belakang :
  4
  6
  7
  1
>>>
```

Muhibah Fata Tika
L200170156
D
Praktikum ALGOSTRUK
Modul 3