

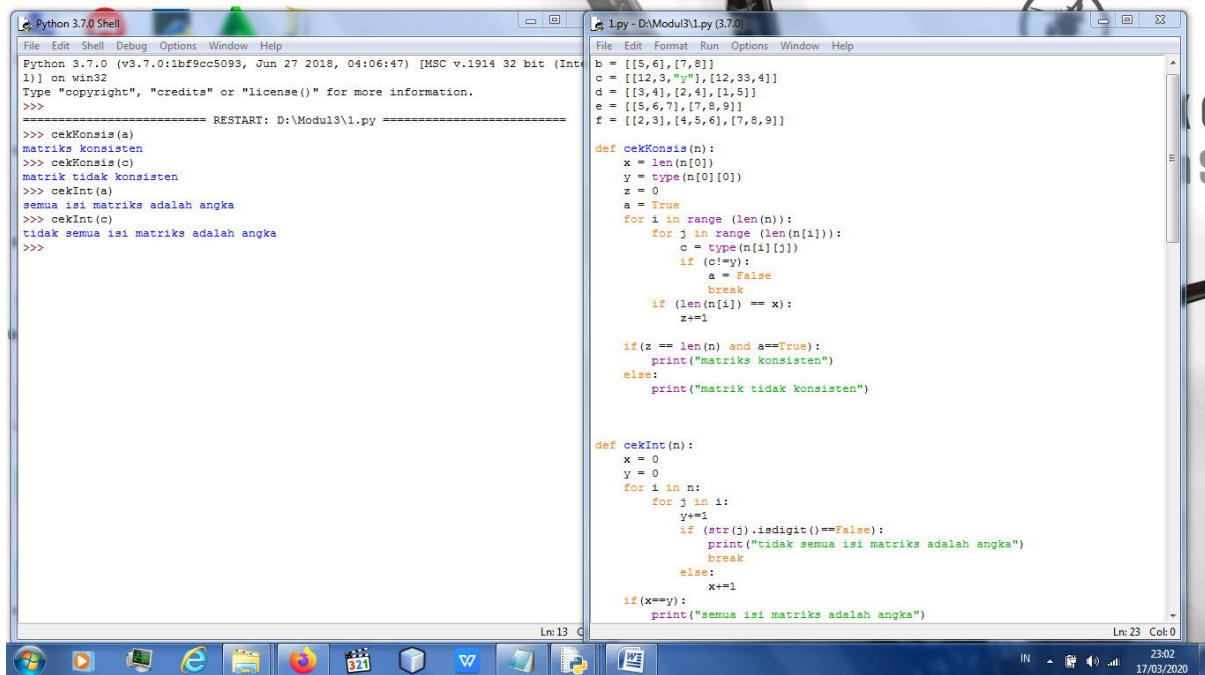
Nama : Amartya Maulana

NIM : L200180196

Kelas : G

Tugas 3

1.



The screenshot shows two windows. The left window is a Python 3.7.0 Shell with the following code and output:

```
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:\Modul3\1.py =====
>>> cekKonsis(a)
matriks konsisten
>>> cekKonsis(c)
matriks tidak konsisten
>>> cekInt(a)
semua isi matriks adalah angka
>>> cekInt(c)
tidak semua isi matriks adalah angka
>>>
```

The right window is a Python 3.7.0 IDLE editor with the following code:

```
1.py - D:\Modul3\1.py (3.7.0)
File Edit Format Run Options Window Help

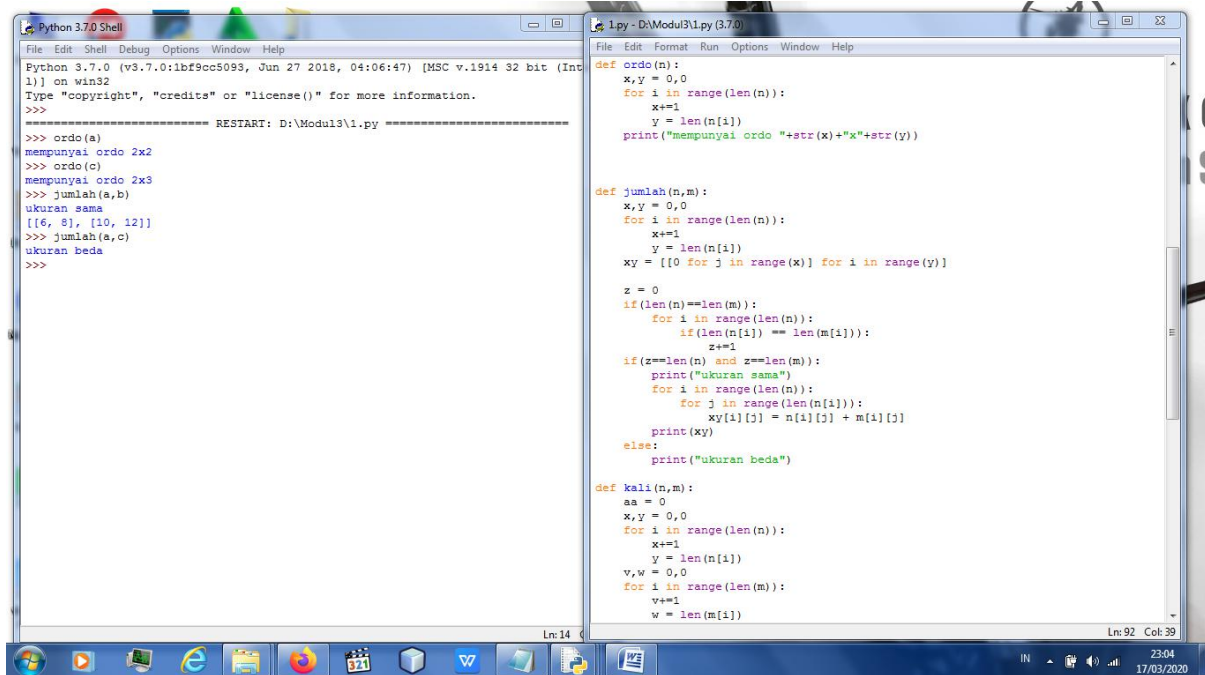
b = [[5,6],[7,8]]
c = [[12,3,"y"],[12,33,4]]
d = [[3,4],[2,4],[1,5]]
e = [[5,6,7],[7,8,9]]
f = [[2,3],[4,5,6],[7,8,9]]

def cekKonsis(n):
    x = len(n[0])
    y = type(n[0][0])
    z = 0
    a = True
    for i in range(len(n)):
        for j in range(len(n[i])):
            c = type(n[i][j])
            if (c!=y):
                a = False
                break
            if (len(n[i]) != x):
                z+=1

    if (z == len(n) and a==True):
        print("matriks konsisten")
    else:
        print("matriks tidak konsisten")

def cekInt(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")
```



The screenshot shows two windows. The left window is a Python 3.7.0 Shell with the following code and output:

```
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:\Modul3\1.py =====
>>> ordo(a)
mempunyai ordo 2x2
>>> ordo(c)
mempunyai ordo 2x3
>>> jumlah(a,b)
ukuran sama
[[6, 8], [10, 12]]
>>> jumlah(a,c)
ukuran beda
>>>
```

The right window is a Python 3.7.0 IDLE editor with the following code:

```
1.py - D:\Modul3\1.py (3.7.0)
File Edit Format Run Options Window Help

def ordo(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))

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")

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])
```

```
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:\Modul3\1.py =====
>>> kali(a,b)
bisa dikalikan
[[0, 0], [0, 0]]
>>> kali(s,e)
bisa dikalikan
[[19, 22], [43, 50]]
>>> kali(a,e)
bisa dikalikan
[[0, 0, 0], [0, 0, 0]]
[[19, 22, 25], [43, 50, 57]]
>>> g=[[1,2,3],[1,2,3]]
>>> kali(a,g)
bisa dikalikan
[[0, 0, 0], [0, 0, 0]]
[[3, 6, 9], [7, 14, 21]]
>>> determHitung(a)
-2
>>> determHitung(c)
'tidak bisa dihitung determinan, bukan matriks bujursangkar'
>>>

D:\Modul3\1.py
Ln: 22 Col: 4
```

```
D:\Modul3\1.py (3.7.0)
File Edit Format Run Options Window Help
kali(n,m):
a = 0
k,y = 0,0
for i in range(len(n)):
    x+=1
    y = len(n[i])
    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)]
        print(vwxy)
        for i in range(len(n)):
            for j in range(len(m[0])):
                for k in range(len(m)):
                    vwxy[i][j] += n[i][k] * m[k][j]

        print(vwxy)
    else:
        print("tidak memenuhi syarat")

determHitung(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:
```

2.

```
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:\Modul3\2.py =====
>>> buatNol(3)
membuat matriks 0 dengan ordo 3x3
[[0, 0, 0], [0, 0, 0], [0, 0, 0]]
>>> buatNol(3,4)
membuat matriks 0 dengan ordo 3x4
[[0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0]]
>>> buatIden(3)
membuat matriks identitas dengan ordo 3x3
[[1, 0, 0], [0, 1, 0], [0, 0, 1]]
>>>

D:\Modul3\2.py
Ln: 22 Col: 4
```

```
D:\Modul3\2.py (3.7.0)
File Edit Format Run Options Window Help
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)])

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)])
```

3.

```
3.py - D:\Modul3\3.py (3.7.0)
File Edit Format Run Options Window Help

class Node:
    def __init__(self, data):
        self.data = data
        self.next = None
class LinkedList:
    def __init__(self):
        self.head = None
    def addF(self, new_data):
        new_node = Node(new_data)
        new_node.next = self.head
        self.head = new_node
    def addE(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)
    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
            node.next = prev.next
            prev.next = node
    def deleteNode(self, position):
        if self.head == None:
            return
        temp = self.head
```

```
3.py - D:\Modul3\3.py (3.7.0)
File Edit Format Run Options Window Help

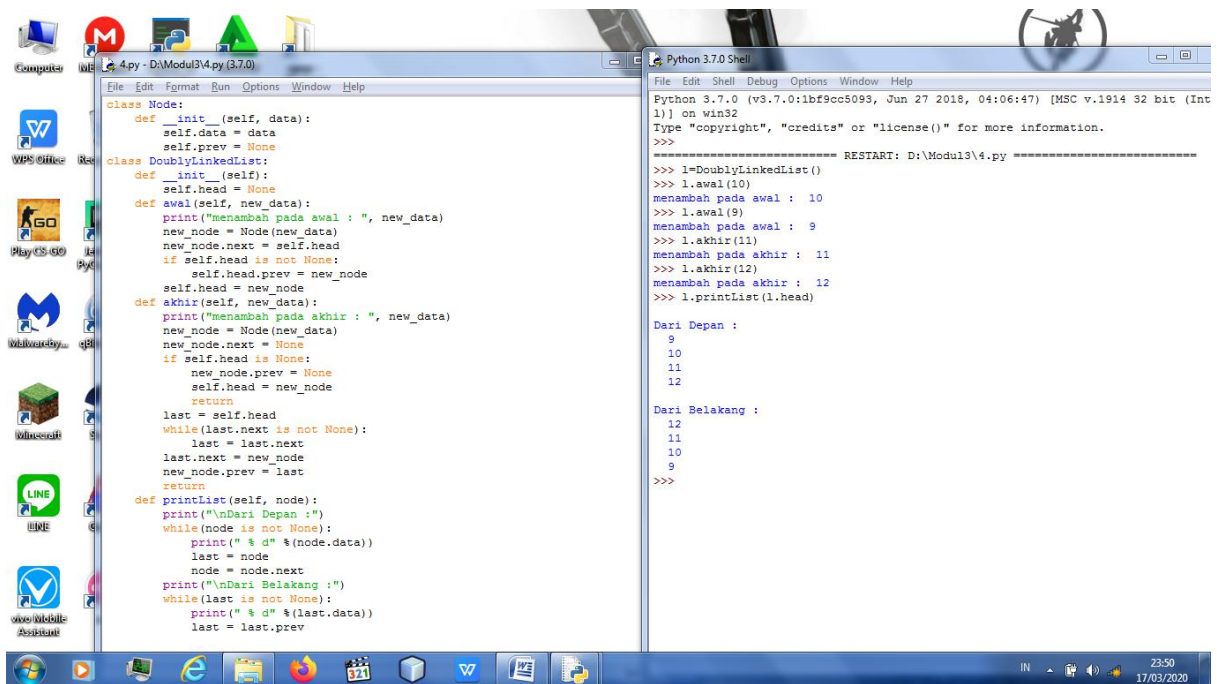
        current_pos +=1
        node.next = prev.next
        prev.next = node
    def deleteNode(self, position):
        if self.head == None:
            return
        temp = self.head
        if position == 0:
            self.head = temp.next
            temp = None
            return
        for i in range(position):
            prev = temp
            temp = temp.next
            if temp is None:
                break
        if temp is None:
            return
        if temp.next is None:
            return
        prev.next = temp.next
        temp = None

    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

Python 3.7.0 Shell
File Edit Shell Debug Options Window Help

Python 3.7.0 (v3.7.0:1bf9ec5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:\Modul3\3.py =====
>>> l=LinkedList()
>>> l.addF(10)
>>> l.addF(20)
>>> l.addF(30)
>>> l.addF(40)
>>> l.addF(50)
>>> l.addF(60)
>>> l.addF(70)
>>> l.addE(0)
>>> l.addE(-10)
>>> l.display()
70 60 50 40 30 20 10 0 -10
>>> l.deleteNode(4)
>>> l.display()
70 60 50 40 20 10 0 -10
>>> l.insert(30,4)
>>> l.display()
70 60 50 40 30 20 10 0 -10
>>> print(l.search(10))
True
>>> print(l.search(100))
False
>>>
```

4.



```
class Node:
    def __init__(self, data):
        self.data = data
        self.prev = None
        self.next = None

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
```

```
Python 3.7.0 Shell
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 bit (Int
1)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:\Modul3\4.py =====
>>> l=DoublyLinkedList()
>>> l.awal(10)
menambah pada awal : 10
>>> l.awal(9)
menambah pada awal : 9
>>> l.akhir(11)
menambah pada akhir : 11
>>> l.akhir(12)
menambah pada akhir : 12
>>> l.printList(l.head)

Dari Depan :
9
10
11
12

Dari Belakang :
12
11
10
9
>>>
```