

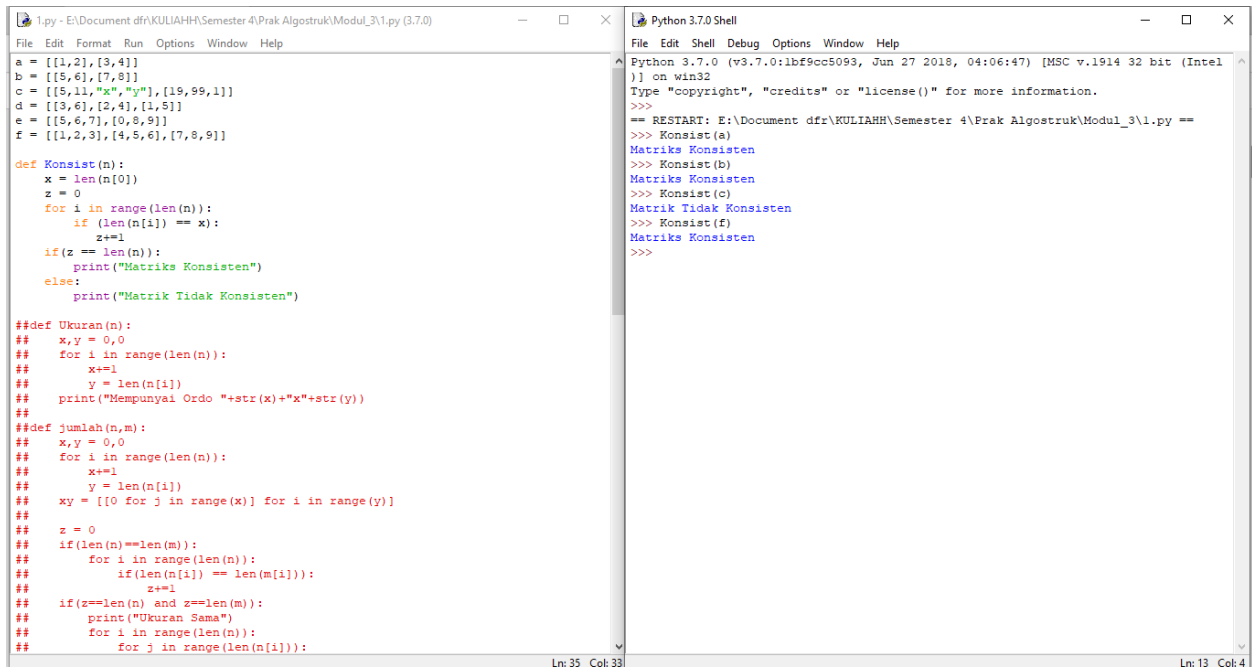
Nama : Diah Fitri Ramadhani

NIM : L200180106

Kelas : D

Tugas Modul 3

No.1a



```
1.py - E:\Document dfr\KULIAHH\Semester 4\Prak Algostruk\Modul_3\1.py (3.7.0)
File Edit Format Run Options Window Help

a = [[1,2],[3,4]]
b = [[5,6],[7,8]]
c = [[5,11,"x","y"],[19,99,1]]
d = [[3,6],[2,4],[1,5]]
e = [[5,6,7],[0,8,9]]
f = [[1,2,3],[4,5,6],[7,8,9]]

def Konsist(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")

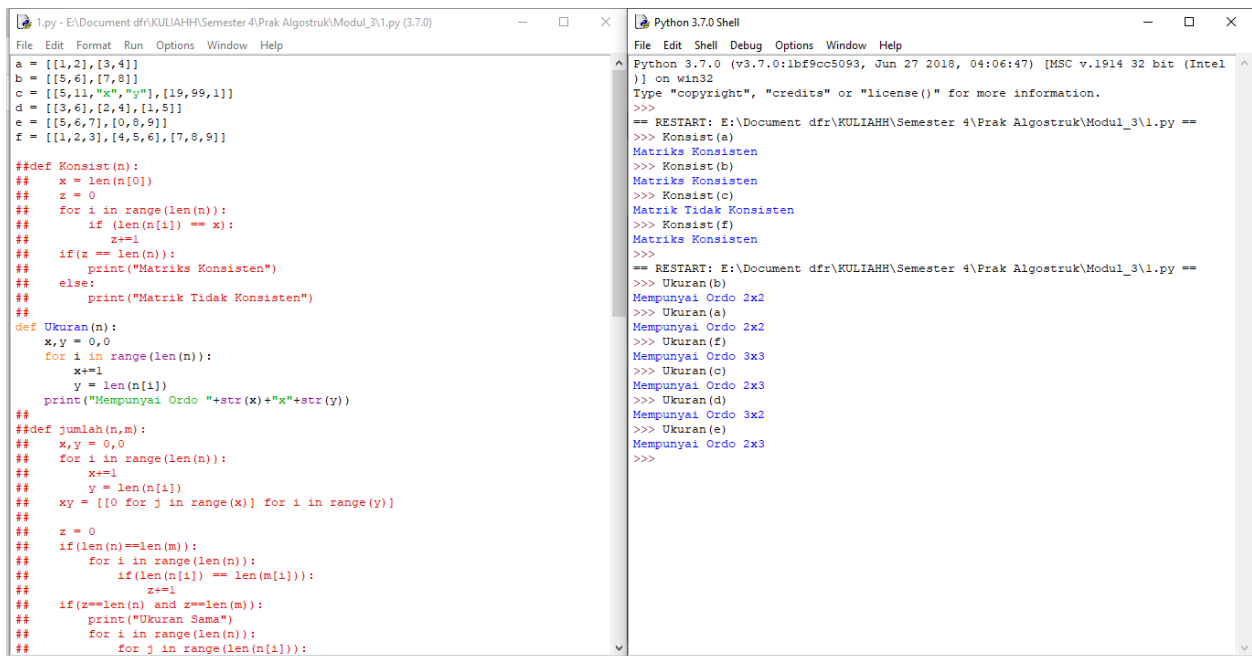
##def Ukuran(n):
##    x,y = 0,0
##    for i in range(len(n)):
##        x+=1
##        y = len(n[i])
##    print("Mempunyai Ordo "+str(x)+"x"+str(y))
##

##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])):

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: E:\Document dfr\KULIAHH\Semester 4\Prak Algostruk\Modul_3\1.py ==
>>> Konsist(a)
Matriks Konsisten
>>> Konsist(b)
Matriks Konsisten
>>> Konsist(c)
Matriks Konsisten
>>> Konsist(d)
Matriks Konsisten
>>> Konsist(e)
Matriks Konsisten
>>> Konsist(f)
Matriks Konsisten
>>>
```

1b



```
1.py - E:\Document dfr\KULIAHH\Semester 4\Prak Algostruk\Modul_3\1.py (3.7.0)
File Edit Format Run Options Window Help

a = [[1,2],[3,4]]
b = [[5,6],[7,8]]
c = [[5,11,"x","y"],[19,99,1]]
d = [[3,6],[2,4],[1,5]]
e = [[5,6,7],[0,8,9]]
f = [[1,2,3],[4,5,6],[7,8,9]]

##def Konsist(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")
##

def Ukuran(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])):

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: E:\Document dfr\KULIAHH\Semester 4\Prak Algostruk\Modul_3\1.py ==
>>> Konsist(a)
Matriks Konsisten
>>> Konsist(b)
Matriks Konsisten
>>> Konsist(c)
Matriks Konsisten
>>> Konsist(d)
Matriks Konsisten
>>> Konsist(e)
Matriks Konsisten
>>> Konsist(f)
Matriks Konsisten
>>>
== RESTART: E:\Document dfr\KULIAHH\Semester 4\Prak Algostruk\Modul_3\1.py ==
>>> Ukuran(b)
Mempunyai Ordo 2x2
>>> Ukuran(a)
Mempunyai Ordo 2x2
>>> Ukuran(f)
Mempunyai Ordo 3x3
>>> Ukuran(c)
Mempunyai Ordo 2x3
>>> Ukuran(d)
Mempunyai Ordo 3x2
>>> Ukuran(e)
Mempunyai Ordo 2x3
>>>
```

1c

```

1.py - E:\Document dfr\KULIAHH\Semester 4\Prak Algoritma\Modul_3\1.py (3.7.0)
File Edit Format Run Options Window Help

## 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")
##
##def Ukuran(n):
##    x,y = 0,0
##    for i in range(len(n)):
##        x+=1
##        y = len(n[i])
##        print("Memunyai 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
##
Ln: 31 Col: 0

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: E:\Document dfr\KULIAHH\Semester 4\Prak Algoritma\Modul_3\1.py ==
>>> jumlah(a,b)
Ukuran Sama
[[6, 8], [10, 12]]
>>> jumlah(a,c)
Ukuran Beda
>>> jumlah(d,f)
Ukuran Beda
>>>

```

1d

```

1.py - E:\Document dfr\KULIAHH\Semester 4\Prak Algoritma\Modul_3\1.py (3.7.0)
File Edit Format Run Options Window Help

# 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])

    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")

Ln: 38 Col: 0

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: E:\Document dfr\KULIAHH\Semester 4\Prak Algoritma\Modul_3\1.py ==
>>> jumlah(a,b)
Ukuran Sama
[[6, 8], [10, 12]]
>>> jumlah(a,c)
Ukuran Beda
>>> jumlah(d,f)
Ukuran Beda
>>> kali(a,b)
Bisa dikalikan
[[19, 22], [43, 50]]
>>> kali(d,f)
Tidak memenuhi syarat
>>> kali(e,f)
Bisa dikalikan
[[78, 96, 114], [95, 112, 129]]
>>>

```

1d

```
1.py - E:\Document dfr\KULIAHH\Semester 4\Prak Algostruk\Modul_3\1.py (3.7.0)
File Edit Format Run Options Window Help
## if(y==x):
##     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")

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 Determinan, bukan Matrix Bujursangkar"
    else:
        return "Tidak Bisa dihitung Determinan, bukan Matrix Bujursangkar"
    return total

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: E:\Document dfr\KULIAHH\Semester 4\Prak Algostruk\Modul_3\1.py ==
>>> determinanHitung(a)
-2
>>> determinanHitung(b)
-2
>>> determinanHitung(c)
'Tidak Bisa dihitung Determinan, bukan Matrix Bujursangkar'
>>> |
```

No.2a

```
2.py - E:\Document dfr\KULIAHH\Semester 4\Prak Algostruk\Modul_3\2.py (3.7.0)
File Edit Format Run Options Window Help
def buatNol(m,n=None):
    if(n==None):
        n=m
    print("Membuat Matriks 0 dengan Ordo "+str(m)+"x"+str(n))
    print([0 for j in range(n)] for i in range(m))

##def buatIdentitas(m):
##     print("Membuat Matriks Identitas dengan Ordo "+str(m)+"x"+str(m))
##     print([1 if j==i else 0 for j in range(m)] for i in range(m))

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: E:\Document dfr\KULIAHH\Semester 4\Prak Algostruk\Modul_3\2.py ==
>>> buatNol(3,3)
Membuat Matriks 0 dengan Ordo 3x3
[[0, 0, 0], [0, 0, 0], [0, 0, 0]]
>>> buatNol(4)
Membuat Matriks 0 dengan Ordo 4x4
[[0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0]]
>>> |
```

2b

```
2.py - E:\Document dfr\KULIAHH\Semester 4\Prak Algotruk\Modul_3\2.py (3.7.0)
File Edit Format Run Options Window Help

##def buatNol(m,n=None):
##     if(n==None):
##         n=m
##     print("Membuat Matriks 0 dengan Ordo "+str(m)+"x"+str(n))
##     print([[0 for j in range(n)] for i in range(m)])

def buatIdentitas(m):
    print("Membuat Matriks Identitas dengan Ordo"+str(m)+"x"+str(m))
    print([[1 if j==i else 0 for j in range(m)] for i in range(m)])

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: E:\Document dfr\KULIAHH\Semester 4\Prak Algotruk\Modul_3\2.py ==
>>> buatNol(3,3)
Membuat Matriks 0 dengan Ordo 3x3
[[0, 0, 0], [0, 0, 0], [0, 0, 0]]
>>> buatNol(4)
Membuat Matriks 0 dengan Ordo 4x4
[[0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0]]
>>>
== RESTART: E:\Document dfr\KULIAHH\Semester 4\Prak Algotruk\Modul_3\2.py ==
>>> buatIdentitas(2)
Membuat Matriks Identitas dengan Ordo2x2
[[1, 0], [0, 1]]
>>> buatIdentitas(6)
Membuat Matriks Identitas dengan Ordo6x6
[[1, 0, 0, 0, 0, 0], [0, 1, 0, 0, 0, 0], [0, 0, 1, 0, 0, 0], [0, 0, 0, 1, 0, 0], [0, 0, 0, 0, 1, 0], [0, 0, 0, 0, 0, 1]]
>>>
```

No.3

```
3.py - E:\Document dfr\KULIAHH\Semester 4\Prak Algotruk\Modul_3\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 pushNw(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
        return self.head
    def deleteNode(self, position):
        if self.head == None:
```

```
3.py - E:\Document dfr\KULIAHH\Semester 4\Prak Algostruk\Modul_3\3.py (3.7.0)
File Edit Format Run Options Window Help

    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)

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: E:\Document dfr\KULIAHH\Semester 4\Prak Algostruk\Modul_3\3.py ==
>>> Llist.insert(1,5)
< main.Node object at 0x02F7E850>
>>> print(Llist.search(17))
False
>>> Llist.display()
2 14 12 22 21 1 1 9
>>> print(Llist.search(22))
True
>>> print(Llist.search(17))
False
>>> Llist.display()
2 14 12 22 21 1 1 9
>>> |

Ln: 10 Col: 0
Ln: 17 Col: 4
```

No.4

```
4.py - E:\Document dfr\KULIAHH\Semester 4\Prak Algostruk\Modul_3\4.py (3.7.0)
File Edit Format Run Options Window Help

class Node:
    def __init__(self, data):
        self.data = data
        self.prev = 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
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: E:\Document dfr\KULIAHH\Semester 4\Prak Algostruk\Modul_3\4.py ==
>>> llist = DoublyLinkedList()
>>> llist.awal(5)
Menambah pada Awal 5
>>> llist.awal(2)
Menambah pada Awal 2
>>> llist.akhir(7)
Menambah pada Akhir 7
>>> llist.akhir(3)
Menambah pada Akhir 3
>>> llist.akhir(1)
Menambah pada Akhir 1
>>> llist.printList(llist.head)

Dari Depan :
2
5
7
3
1

Dari Belakang :
1
3
7
5
2
>>> |

Ln: 5 Col: 0
Ln: 31 Col: 4
```