Nama : Fawwaz Haidar A.K
NIM   : L200183143
Kelas : H

Modul 3

Praktikum Algoritma dan Struktur Data

1.

   a. Konsisten isi dan ukuran matriks

```python
"""modul 3"""

A = [[2,3],
     [3,1]]
B = [[6,3],
     [4,2]]

a = len(A)
b = len(B)

#1
#konsistensi isi dan ukuran matriks
N = 5
M = 4

res = [ [ 0 for i in range(N) ] for j in range(M) ]

print("The matrix after initializing : " + str(res))
```

   b. Mengambil ukuran matriks

```python
#ukuran matriks
res = [sum(len(row) > idx for row in B)
       for idx in range(max(map(len, B)))]

print ("The size of matrix : " + str(res))
```

   c. Menjumlahkan dua matriks

```python
#menjumlahkan dua matriks
for x in range(0, len(A)):
    for y in range(0, len(A[0])):
        print (A[x][y] + B[x][y], end=' ')
    print ()

print (' ')
```

   d. Mengalikan dua matriks

```python
#mengalikan dua matriks
X = []

for x in range(0, len(A)):
    row = []
    for y in range(0, len(A[0])):
        total = 0
        for z in range(0, len(A)):
            total = total + (A[x][z] * B[z][y])
        row.append(total)
    X.append(row)

for x in range(0, len(X)):
    for y in range(0, len(X[0])):
        print (X[x][y], end=' ')
    print ()

print (' ')
```

e. Menghitung determinan matriks

```python
#menghitung determinan matriks
def determinantOfMatrix(A,n):
    temp = [0]*n
    total=1
    det=1

    for i in range(0,n):
        index=i

        while(A[index][i] == 0 and index < n):
            index+=1

        if(index == n):
            continue

        if(index != i):
            for j in range(0,n):
                A[index][j],A[i][j] = A[i][j],A[index][j]
            det = det*int(pow(-1,index-i))

        for j in range(0,n):
            temp[j] = A[i][j]

        for j in range(i+1,n):
            num1 = temp[i]
            num2 = A[j][i]

            for k in range(0,n):

                A[j][k] = (num1*A[j][k]) - (num2*temp[k])

            total = total * num1

    for i in range(0,n):
        det = det*A[i][i]


    return int(det/total)

print("Determinant of the matrix is : ",determinantOfMatrix(A,a))
```

2.

    a. Membangkitkan matriks berisi nol semua

```python
#2
#membangkitkan matrix 0
def buatNol(m):
    print ([[0 for j in range(m)] for i in range(m)])
```

    b. Membangkitkan matriks identitas

```python
#membangkitkan matrix identitas
def buatIdentitas(size):
    for row in range(0, size):
        for col in range(0, size):

            # Here end is used to stay in same line
            if (row == col):
                print("1 ", end=" ")
            else:
                print("0 ", end=" ")
        print()
```

3.

    a. Mencari data yg isinya tertentu

```python
#3. Linked List
#mencari data isinya tertentu
class node:
    def __init__(self, next=None, data=None):
        self.next = next
        self.data = data
    def getdata(self):
        return self.data
    def setnext(self, newNext):
        self.next = newNext
    def recSearch(node, l, r, x):
        if r < l:
            return -1
        if node[l] == x:
            return l
        if node[r] == x:
            return r
        return LinkedList.recSearch(node, l+1, r-1, x)
```

    b. Menambah suatu simpul diawal

```python
#menambah suatu simpul diawal
    def tambahDepan(self, i):
        self.i = i
        node.append(i)
```

    c. Menambah suatu simpul diakhir

```python
#menambah suatu simpul diakhir
    def tambahAkhir(self, i):
        self.i = i
        node.prepend(i)
```

d. Menyisipkan suatu simpul dimana saja

```python
#menyisipkan simpul dimana saja
class LinkedList:
    def __init__(self, head=None):
        self.head = head
    def tambah(self, prev, baru):
        baru.next = prev.next
        prev.next = baru
```

e. Menghapus suatu simpul dimana saja

```python
#menghapus simpul dimana saja
    def hapus(self, item):
        current = self.head
        previous = None
        found = False
        while current != None and not found:
            if current.getData() == item:
                found = True
                print(item, "Ditemukan")
            else:
                previous = current
                current = current.getNext()
        if found == False:
            print(item, "tidak Ditemukan")
        elif previous == None:
            self.head = current.getNext()
        else:
            previous.setNext(current.getNext())
```

4.

a. Mengunjungi dan mencetak tiap simpul dari depan dan belakang

```python
#4. Double Linked List
#mengunjungi dan mencetak data tiap simpul dari depan maupun belakang
    def cetakdepan(self):
        ini = self.head
        while ini is not None:
            print(ini.data)
            ini = ini.next

    def cetakbelakang(self):
        for i in data(len(data),0):
            return i
```

b. Menambah suatu simpul diawal

```python
#menambah suatu simpul diawal
    def tambahDepan(self, i):
        self.i = i
        node.append(i)
```

c. Menambah suatu simpul diakhir

```python
#menambah suatu simpul diawal
    def tambahAkhir(self, i):
        self.i = i
        node.prepend(i)




a = node(2)
b = node(7)
c = node(15)
d = node(28)
e = node(33)
f = node(49)
g = node(56)

a.next = b
b.prev = a
b.next = c
c.prev = b
c.next = d
d.prev = c
d.next = e
e.prev = d
e.next = f
f.prev = e
f.next = g
g.prev = f

node = [2,7,15,28,33,49,56]
```