**PRACTICUM REPORT**

**ALGORITHM AND DATA STRUCTURES**

**MODUL 3 : COLLECTIONS, ARRAYS, AND LINKED**

**STRUCTURES**



**Disusun Oleh :**

**ONIC AGUSTINO**

**L200234275**

**X**

**INFORMATICS ENGINEERING**

**FACULTY OF COMMUNICATION AND INFORMATICS**

**UNIVERSITAS MUHAMMADIYAH SURAKARTA**

**YEARS 2024/2025**

**1.11 Questions**

1.Regarding two-dimensional arrays, we will create a matrix data type

that contains numbers. For that, create functions

• to ensure that the contents and size of the matrix are consistent

(because each member of the outer list may have a different size,

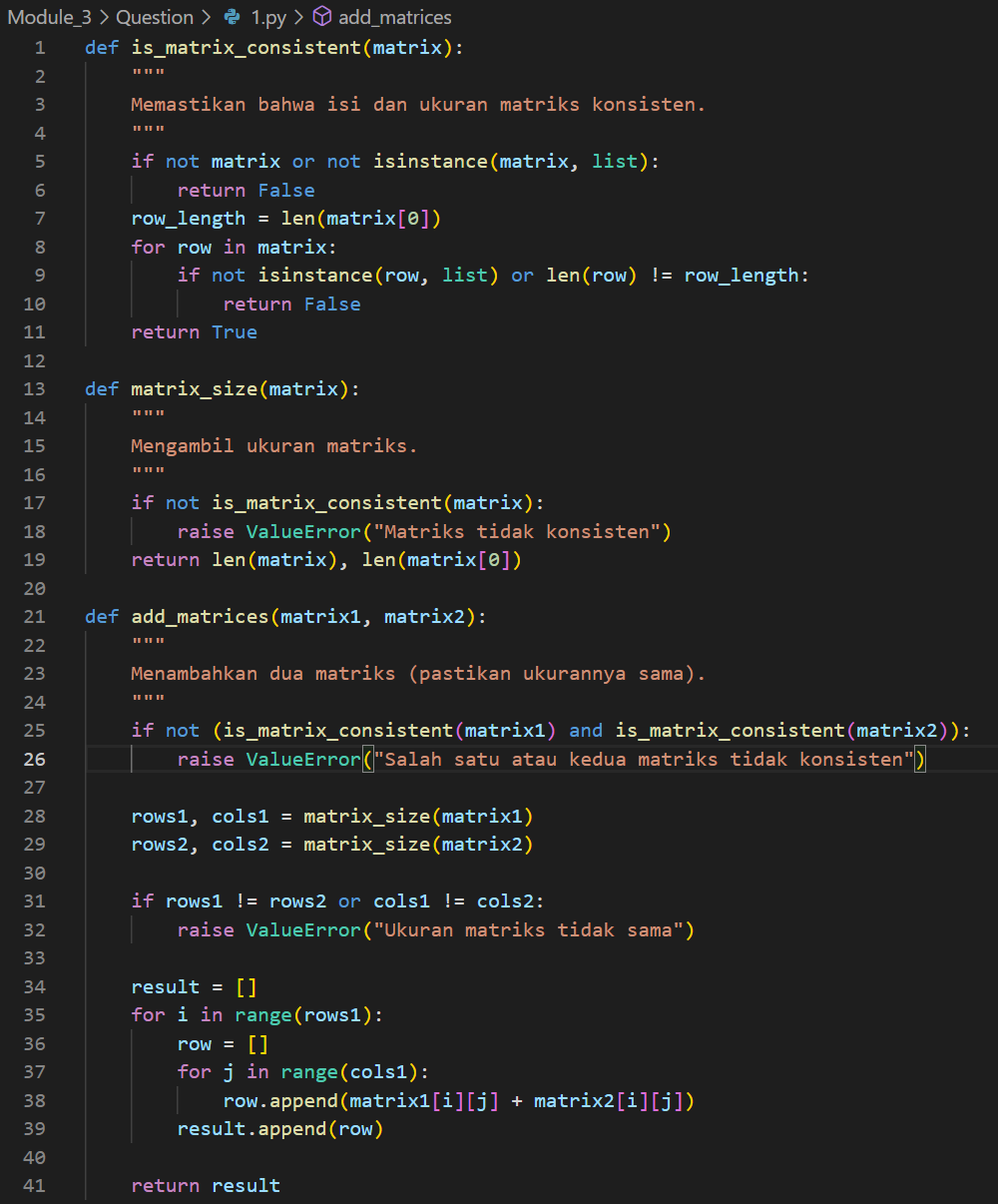
and may even be a different type!),

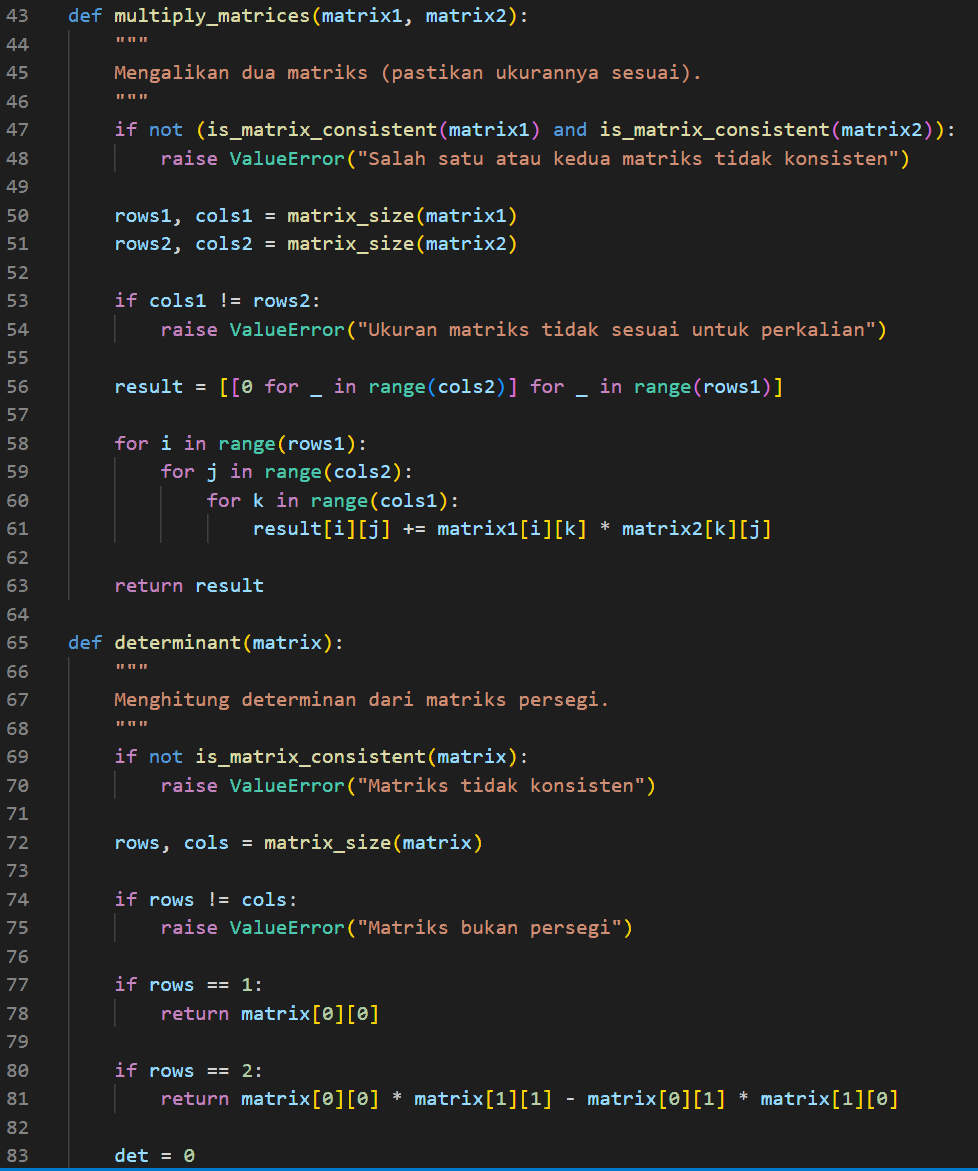
• to take the matrix size,

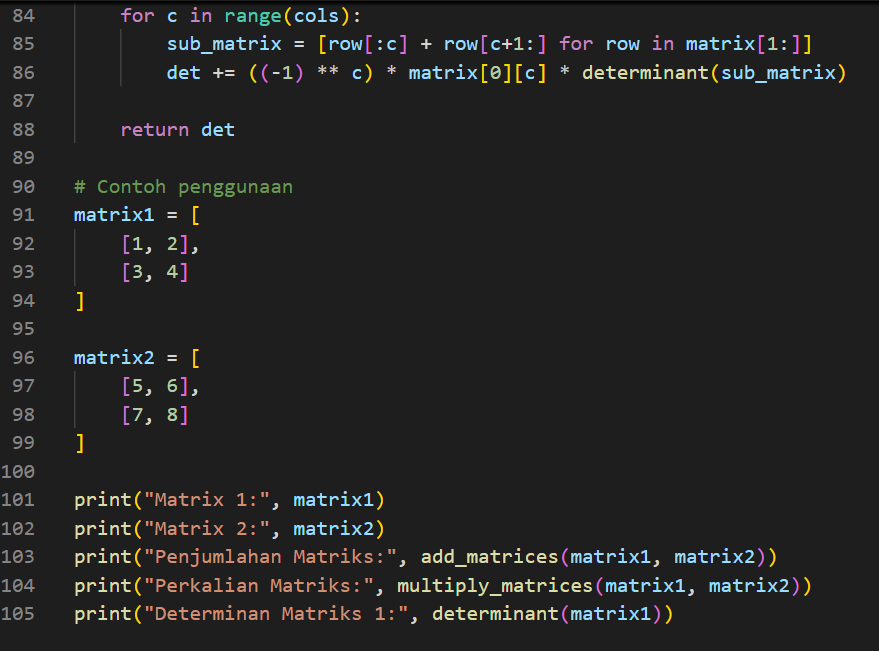
• to add two matrices (make sure the sizes match),

• to multiply two matrices (make sure the sizes match),

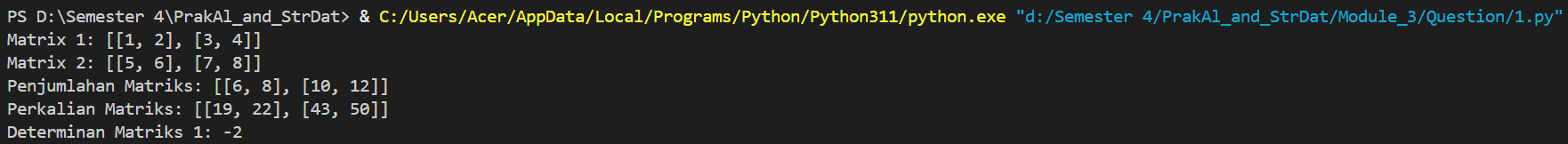
• to calculate the determinant of a square matrix.

**





Picture 1.1 the code 1.py



Picture 1.2 the output 1.py

2. Regarding the matrix and list comprehension, create (using list

comprehension) functions

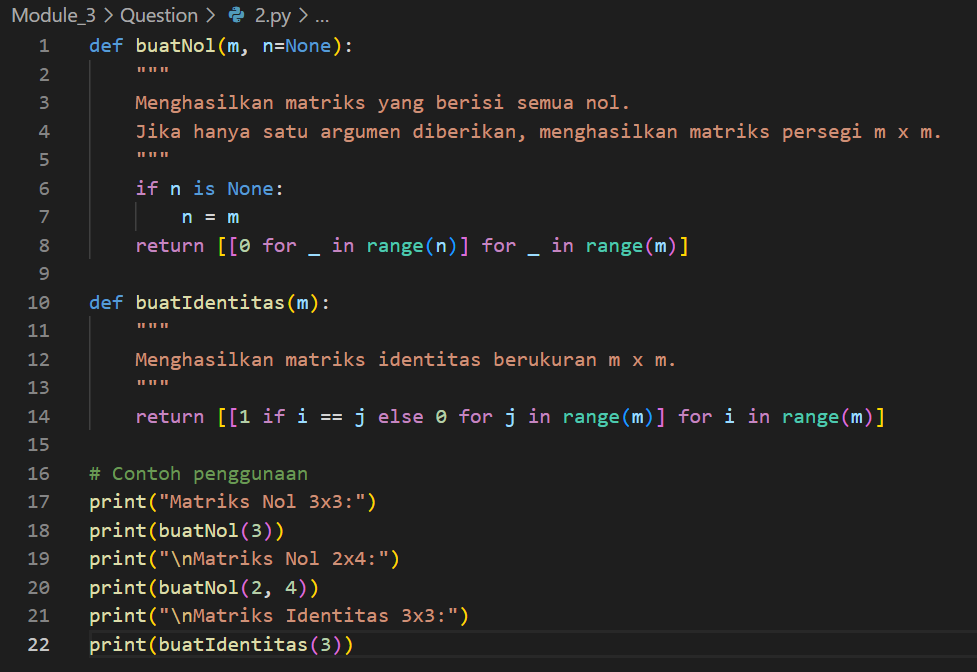
• to generate a matrix containing all zeros, given its size. calling:

buatNol(m,n) and buatNol(m). Calling the latter method will

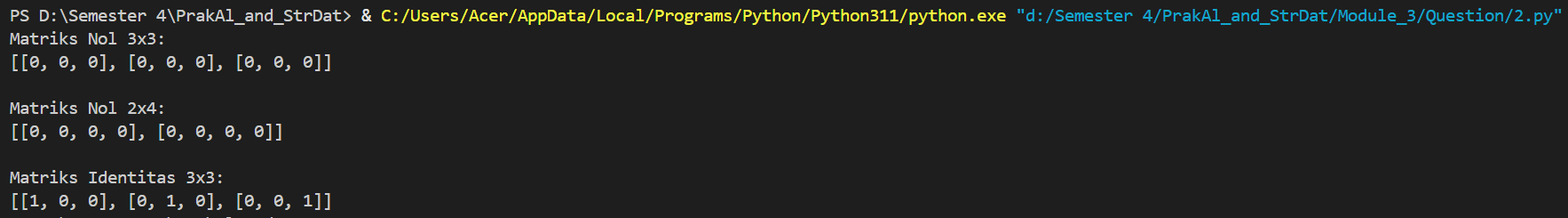
provide a square matrix of sizes m × m.

• to generate an identity matrix, given its size. Call:

buatIdentitas(m.



Picture 2.1 the code 2.py



Picture 2.2 the output 2.py

3. Regarding the linked list, create a function for

• search for data with certain contents: cari(head,yang

dicari)

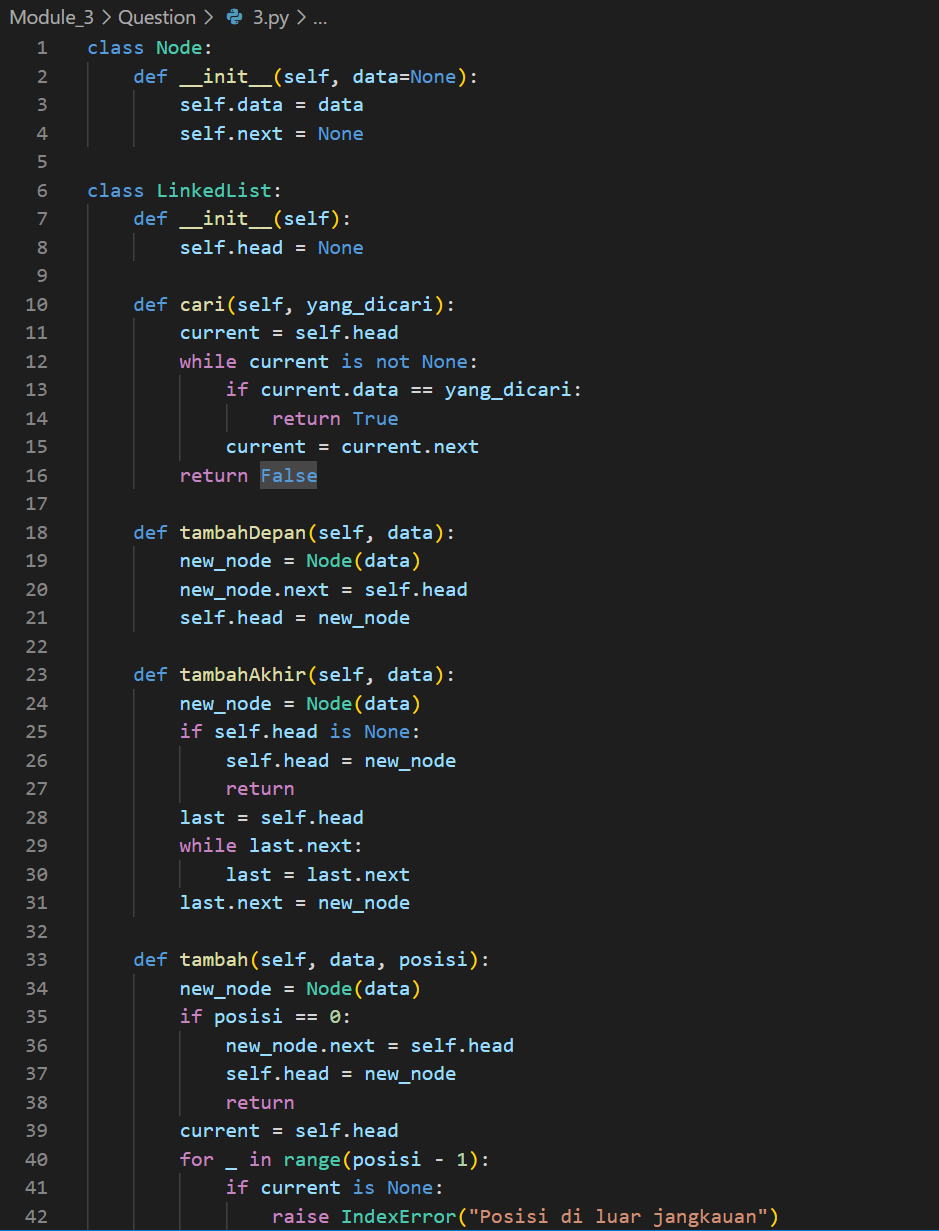
• add a node at the beginning: tambahDepan(head)

• add a node at the end: tambahAkhir(head)

• insert a node anywhere: tambah(head,posisi)

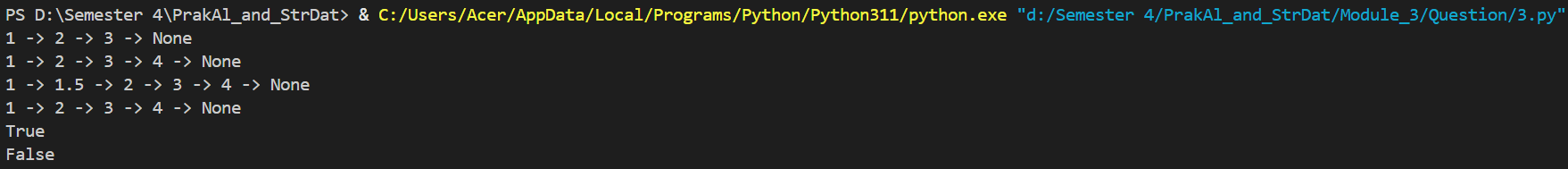
• delete a node at the beginning, at the end, or anywhere:

hapus(posisi)





Picture 3.1 the code 3.py



Picture 3.2 the output 3.py

4. Regarding the doubly linked list, create a function for

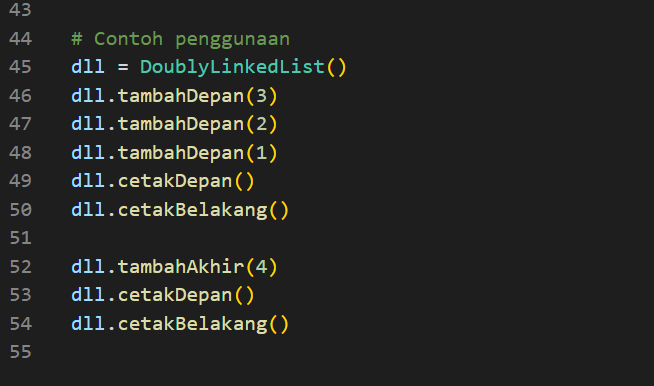
• visit and print data for each node from the front and from the

back.

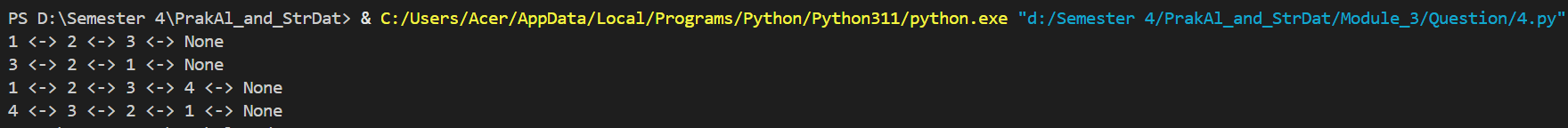
• add a node at the beginning

• add a node at the end





Picture 4.1 the code 4.py



Picture 4.2 the output 4.py