

COLLEGE OF COMPUTING AND INFORMATION SCIENCES

Final Assessment of Lab Exam (Summer 2021 Semester)

KIEI			
Class Id	107285	Course Title	Numerical Computing LAB
Program	BSCS	Campus / Shift	MAIN MORNING
Date	16-07-2021	Total Marks	20
Duration	2.5 hours	Faculty Name	Aziz Mehmoud Farooqi
Student Id	10619	Student Name	Muhammad Umar Khan
Code	A1		

Instructions:

- Fill out your Student ID and Student Name in above header.
- Do not remove or change any part question paper.
- Write down your answers with title "Answer for Question# 00".
- Handwritten text or image should be on A4 size page with clear visibility of contents.
- In case of CHEATING, COPIED material or any unfair means would result in negative marking or ZERO.
- <u>Caution:</u> Duration to perform Final Assessment is <u>02 hours & 30 mins only</u>. If you failed to upload answer sheet on LMS (in PDF format) within 2.5 hours limit, you would be considered as ABSENT/FAILED.

Instructions: Attempt all two questions.

[10 Marks]

Question 1:

Write a Dynamic Python code for solving any set of equations problem (3 unknown variables or 4 etc.) using **Gaussian Elimination Method**.

You have to define two returnable functions named *funForwardEDynamically*(a) and *funBackSubDynamically*(a) that takes only one array matrix and returns an another array.

After creating the functions, first you will call the *funForwardEDynamically* then *funBackSubDynamically* in the main function...

Solve the following two set of equations by Gaussian elimination one by one:

3 variables & 3 equations:

$$\begin{array}{rcl}
-x & -5y & -5z & = & 2 \\
4x & -5y & +4z & = & 19 \\
x & +5y & -z & = & -20
\end{array}$$

4 variables & 4 equations:

CODE: (3 VARIABLES 3 EQUATIONS):

```
def BackDub(a):
def funForwardEDynamically(a):
a3 = a[2][3] / a[2][2]
a2 = ((a[1][3]) + (-a[1][2] * a3)) / (a[1][1])
print('a3 Value is:', a3)
```

```
print('a2 Value is:', a2)
print('a1 Value is:', a1)
import numpy as np
equationTHREE = np.array([[-1,-5,-5,2],[4,-5,4,19],[1,5,-1,-20]])

FORWARD = funDynamic(equationTHREE)

print(FORWARD)
backword = BackDub(FORWARD)
res = backword
res
```

OUTPUT:

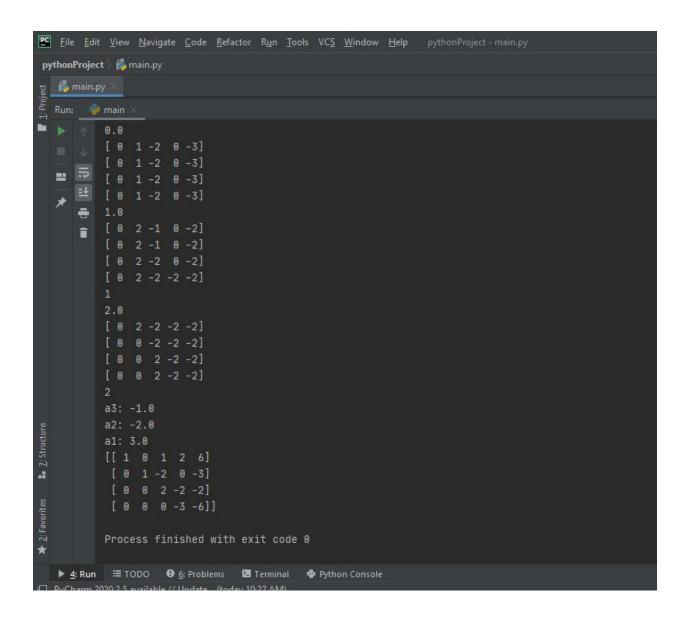
```
File Edit View Navigate Code Refactor Run Tools VCS Window Help pythonProject - main.py
pythonProject ) 👸 main.py
ਲੂ 🐔 main.py 🔻
  Run: w main
C:\Users\Ali\PycharmProjects\Scripts\python.exe C:/Users/Ali/pythonProject/main.py
       4
           [ 0 -5 4 19]
           [ 0 -25 4 19]
           [ 0 -25 -16 19]
           [ 0 -25 -16 27]
           -1.0
           -0.0
           a3 Value is: 3.0
           a2 Value is: -3.0
           al Value is: -2.0
           [ 0 -25 -16 27]
   🕨 4: Run 🔚 TODO 🛭 6: Problems 🚨 Terminal ಿ Python Console
```

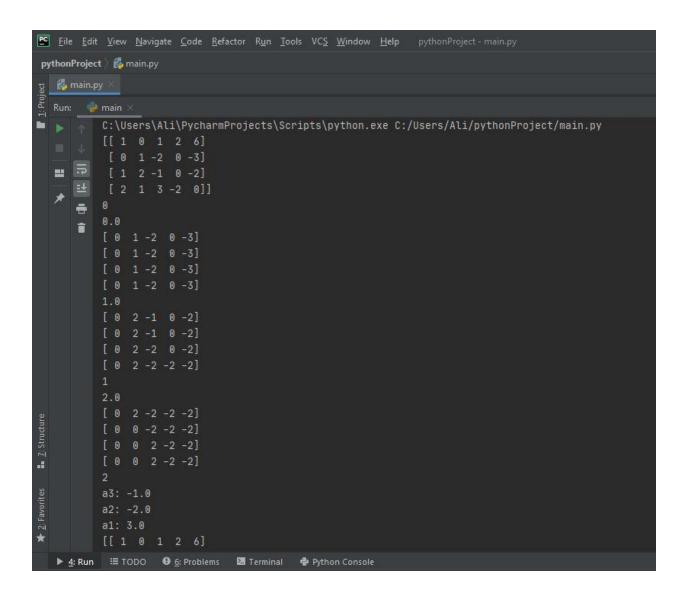
```
<u>File Edit View Navigate Code Refactor Run Tools VCS Window H</u>elp pythonProject - main.py
pythonProject ) 🚜 main.py
  👸 main.py 🗵
          -4.0
          [ 0 -25 4 19]
          [ 0 -25 -16 19]
           [ 0 -25 -16 27]
          -1.0
                5 -1 -20]
           [ 0 0 -1 -20]
           [ 0 0 -6 -20]
                  0 -6 -18]
          -0.0
                 0 -6 -18]
                 0 -6 -18]
           [ 0 0 -6 -18]
          [ 0 0 -6 -18]
          a3 Value is: 3.0
          a2 Value is: -3.0
          al Value is: -2.0
          [[-1 -5 -5 2]
           [ 0 0 -6 -18]]
           Process finished with exit code 0
  ▶ <u>4</u>: Run III TODO <u>9</u> <u>6</u>: Problems III Terminal ♣ Python Console
```

4X4 CODE:

```
mid = mid + (a kvalues * a cellvalues)
import numpy as np
def funForwardEDynamically(a):
a = np.array([[1, 0, 1, 2, 6], [0, 1, -2, 0, -3], [1, 2, -1, 0, -2], [2, 1,
a3 = a[2][3] / a[2][2]
a2 = ((a[1][3]) + (-a[1][2] * a3)) / (a[1][1])
print('a3:', a3)
print('a2:', a2)
eq3 = np.array([[1,0,1,2,6],[0,1,-2,0,-3],[1,2,-1,0,-2],[2,1,3,-2,0]])
forword = funDynamic(eq3)
print(forword)
backword = BackDub(forword)
```

OUTPUT:





Question 2:

Write a Static Python code for solving the particular set of equation problem (4 unknown variables and 4 equations) using **Jacobi Method**.

Solve the following set of equations by Jacobi:

4 variables & 4 equations:

CODE:

```
Main Matrix = [[10, -1, 2, 0, 6], [-1, 11, -1, 3, 25], [2, -1, 10, -1, -11],
Matrix = np.array(Main Matrix, dtype=float)
```

