OS -The Banker's Algorithm Assignment

Name: Mostafa Sami Fayez

ID: 15X0076

Section: 4

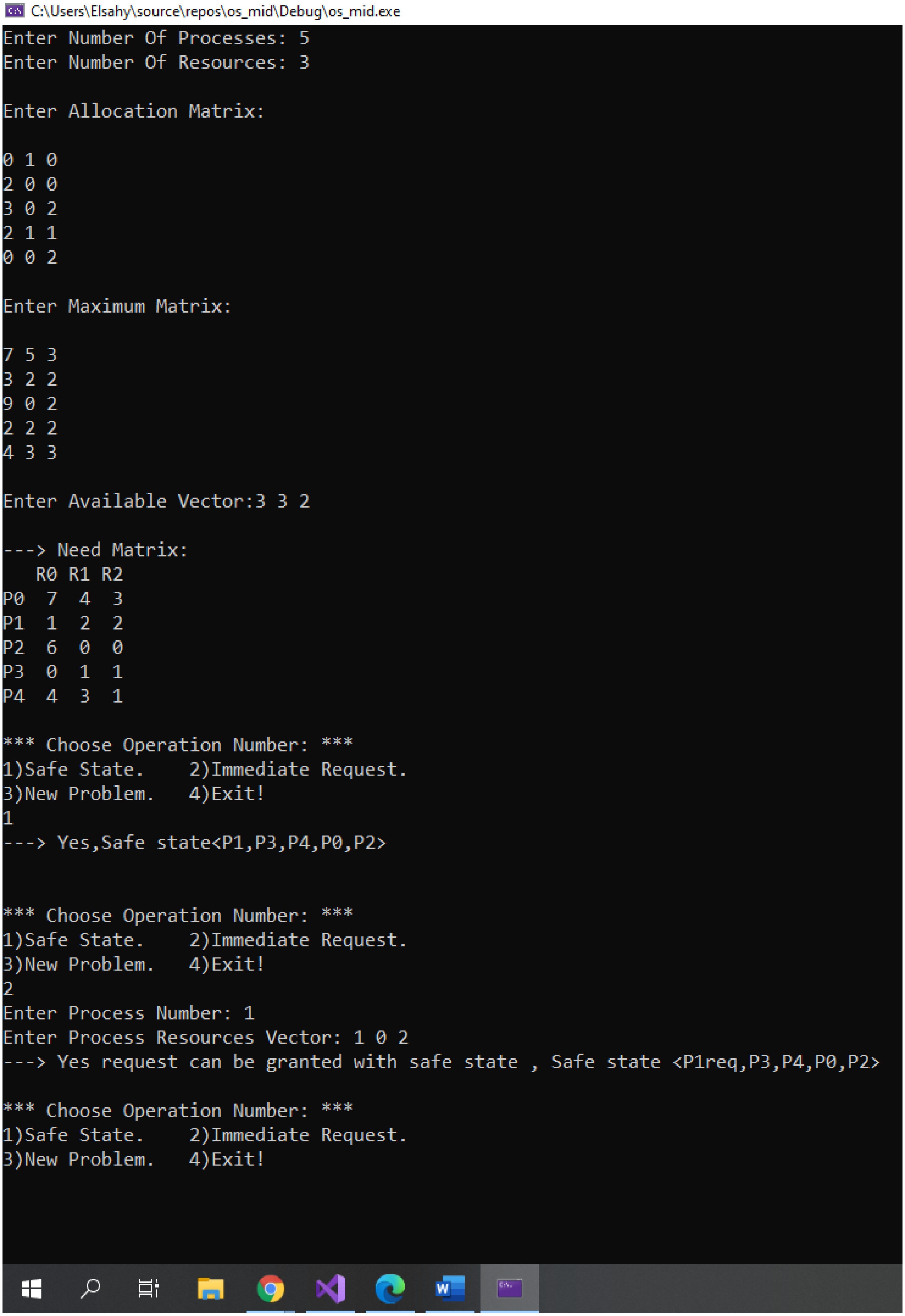
Project LINK

Code Design:

- <u>Code has additional feature that user can ask several questions several times on the same problem or choose to enter another problem or just exit.</u>

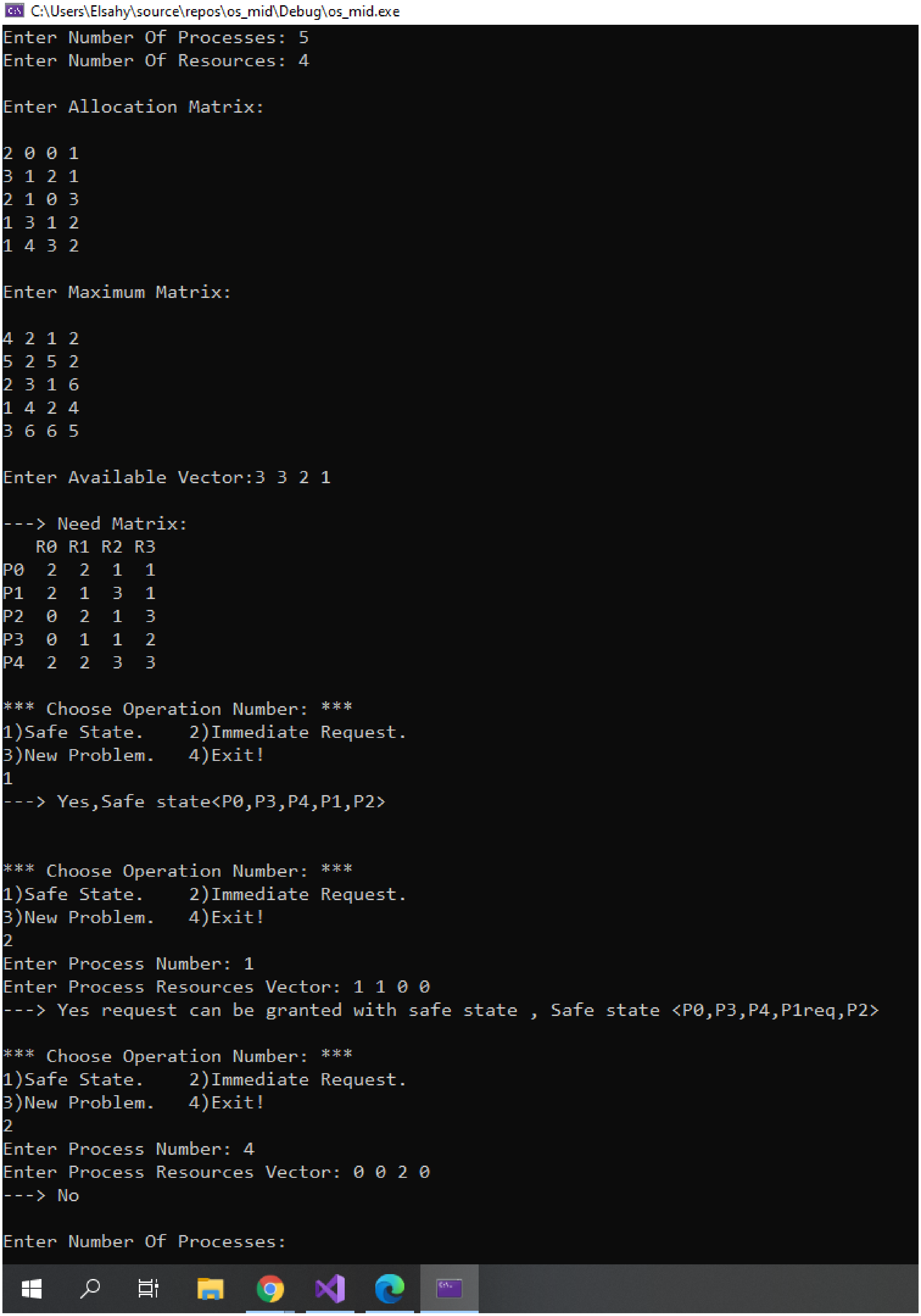
- <u>Implementation Is Around 3 main Blocks Working as described below:</u>

1. -Safe Check Function:
    a. Role: Find if the given table is safe or not
    b. Input: Table (allocation, need, available, flag)
    c. Output: Queue contain process run during the function execution in order
    d. If Queue size equal to number of process, then it is safe.
2. Taking Input and Calculate need matrix.
3. Operation Block which handles user choice and run algorithm out of 4 Choices:
    a. Run Safety Check
        i. Where we call Block 1
    b. Run Immediate Request
        i. Do some logic first to cout no if it is clear.
        ii. Call safety check again here to make sure the request can be done without making dead lock.
    c. Enter New Problem
        i. Branch to Block 3 again
    d. Exit Program



C:\Users\Elsahy\source\repos\os_mid\Debug\os_mid.exe

```
Enter Number Of Processes: 5
Enter Number Of Resources: 3

Enter Allocation Matrix:

0 1 0
2 0 0
3 0 2
2 1 1
0 0 2

Enter Maximum Matrix:

7 5 3
3 2 2
9 0 2
2 2 2
4 3 3

Enter Available Vector:3 3 2

---> Need Matrix:
   R0 R1 R2
P0  7  4  3
P1  1  2  2
P2  6  0  0
P3  0  1  1
P4  4  3  1

*** Choose Operation Number: ***
1)Safe State.    2)Immediate Request.
3)New Problem.   4)Exit!
1
---> Yes,Safe state<P1,P3,P4,P0,P2>


*** Choose Operation Number: ***
1)Safe State.    2)Immediate Request.
3)New Problem.   4)Exit!
2
Enter Process Number: 1
Enter Process Resources Vector: 1 0 2
---> Yes request can be granted with safe state , Safe state <P1req,P3,P4,P0,P2>

*** Choose Operation Number: ***
1)Safe State.    2)Immediate Request.
3)New Problem.   4)Exit!
```



```
Enter Number Of Processes: 5
Enter Number Of Resources: 4

Enter Allocation Matrix:

2 0 0 1
3 1 2 1
2 1 0 3
1 3 1 2
1 4 3 2

Enter Maximum Matrix:

4 2 1 2
5 2 5 2
2 3 1 6
1 4 2 4
3 6 6 5

Enter Available Vector:3 3 2 1

---> Need Matrix:
   R0 R1 R2 R3
P0  2  2  1  1
P1  2  1  3  1
P2  0  2  1  3
P3  0  1  1  2
P4  2  2  3  3

*** Choose Operation Number: ***
1)Safe State.    2)Immediate Request.
3)New Problem.   4)Exit!
1
---> Yes,Safe state<P0,P3,P4,P1,P2>


*** Choose Operation Number: ***
1)Safe State.    2)Immediate Request.
3)New Problem.   4)Exit!
2
Enter Process Number: 1
Enter Process Resources Vector: 1 1 0 0
---> Yes request can be granted with safe state , Safe state <P0,P3,P4,P1req,P2>

*** Choose Operation Number: ***
1)Safe State.    2)Immediate Request.
3)New Problem.   4)Exit!
2
Enter Process Number: 4
Enter Process Resources Vector: 0 0 2 0
---> No

Enter Number Of Processes:
```