Ain Shams University

Faculty of Engineering

Computer and System department

Bankers Algorithm

Submitted by:

Ahmed Mohamed Ahmed Abd El-Hamed Hassan

Code: 1700157

Section: 1

Safe State Tests:

```
Enter num of resources : 3
Enter num of processes : 5

Enter Allocation array for P0 : 0 1 0
Enter Allocation array for P1 : 2 0 0
Enter Allocation array for P2 : 3 0 2
Enter Allocation array for P3 : 2 1 1
Enter Allocation array for P4 : 0 0 2

Enter Max array for P0 : 7 5 3
Enter Max array for P1 : 3 2 2
Enter Max array for P2 : 9 0 2
Enter Max array for P3 : 2 2 2
Enter Max array for P4 : 4 3 3

Enter Available array
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

Do you want to check safety? (1-yes 2-no)
your choice : 1
Yes , Safe state <P1,P3,P4,P0,P2>

Do you want to make request? (1-yes 2-no)
your choice : 2


Do you want to test agian?(y/n)
```

```
Enter num of resources : 3
Enter num of processes : 5

Enter Allocation array for P0 : 0 4 0
Enter Allocation array for P1 : 2 0 0
Enter Allocation array for P2 : 3 0 2
Enter Allocation array for P3 : 2 1 1
Enter Allocation array for P4 : 0 0 2

Enter Max array for P0 : 7 5 3
Enter Max array for P1 : 3 2 2
Enter Max array for P2 : 9 0 2
Enter Max array for P3 : 2 2 2
Enter Max array for P4 : 4 3 3

Enter Available array
3 0 2

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

Do you want to check safety? (1-yes 2-no)
your choice : 1
No

Do you want to make request? (1-yes 2-no)
your choice : 2


Do you want to test agian?(y/n)
```

Requests Tests:

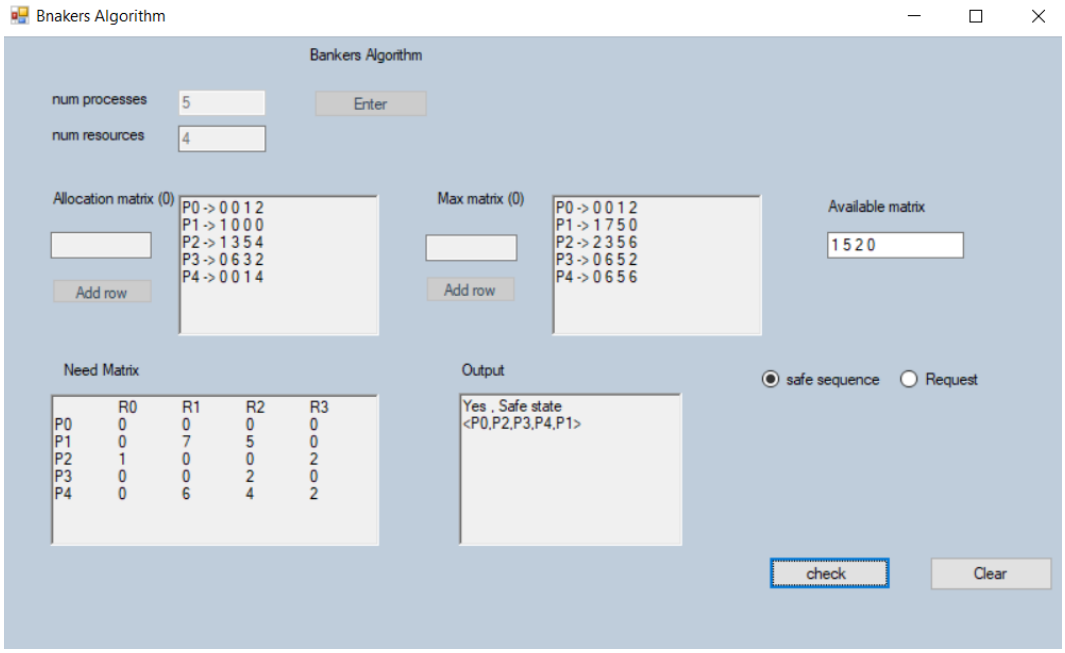
```
Enter num of resources : 3
Enter num of processes : 5

Enter Allocation array for P0 : 0 1 0
Enter Allocation array for P1 : 2 0 0
Enter Allocation array for P2 : 3 0 2
Enter Allocation array for P3 : 2 1 1
Enter Allocation array for P4 : 0 0 2

Enter Max array for P0 : 7 5 3
Enter Max array for P1 : 3 2 2
Enter Max array for P2 : 9 0 2
Enter Max array for P3 : 2 2 2
Enter Max array for P4 : 4 3 3

Enter Available array
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

Do you want to check safety? (1-yes 2-no)
your choice : 1
Yes , Safe state <P1,P3,P4,P0,P2>

Do you want to make request? (1-yes 2-no)
your choice : 1
enter index of process: 1
enter request of process 1 : 1 0 2

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

Yes , Safe state <P1req,P1,P3,P4,P0,P2>

Do you want to test agian?(y/n)
```

```
Enter num of resources : 3
Enter num of processes : 5

Enter Allocation array for P0 : 0 4 0
Enter Allocation array for P1 : 2 0 0
Enter Allocation array for P2 : 3 0 2
Enter Allocation array for P3 : 2 1 1
Enter Allocation array for P4 : 0 0 2

Enter Max array for P0 : 7 5 3
Enter Max array for P1 : 3 2 2
Enter Max array for P2 : 9 0 2
Enter Max array for P3 : 2 2 2
Enter Max array for P4 : 4 3 3

Enter Available array
3 0 2

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

Do you want to check safety? (1-yes 2-no)
your choice : 2


Do you want to make request? (1-yes 2-no)
your choice : 1
enter index of process: 1
enter request of process 1 : 1 0 2
No
```

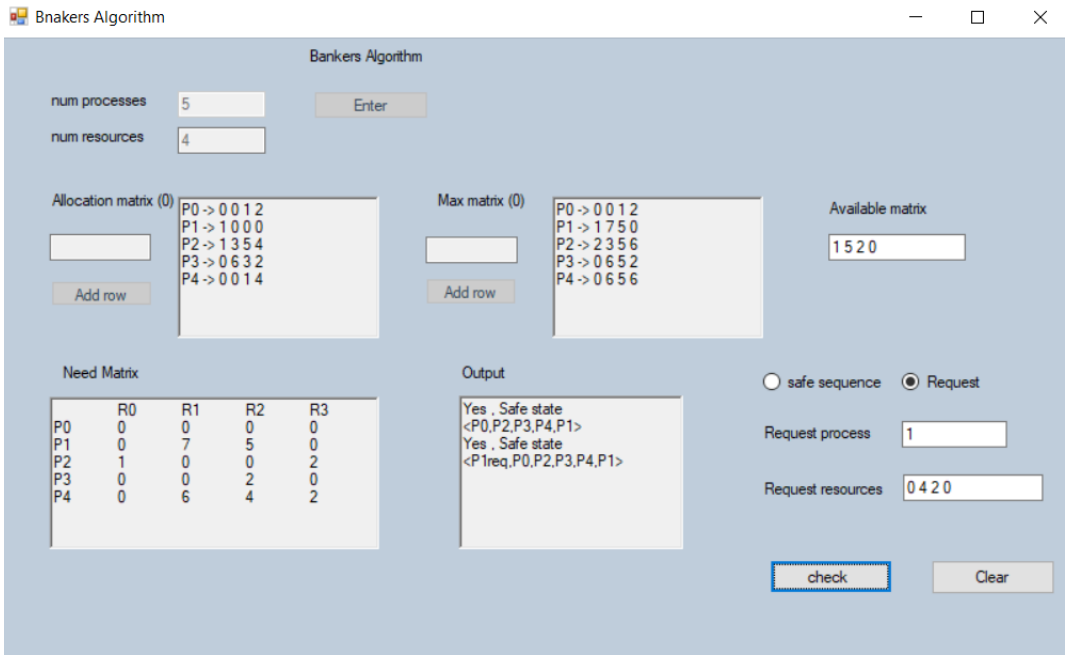
```
Enter num of resources : 4
Enter num of processes : 5

Enter Allocation array for P0 : 0 0 1 2
Enter Allocation array for P1 : 1 0 0 0
Enter Allocation array for P2 : 1 3 5 4
Enter Allocation array for P3 : 0 6 3 2
Enter Allocation array for P4 : 0 0 1 4

Enter Max array for P0 : 0 0 1 2
Enter Max array for P1 : 1 7 5 0
Enter Max array for P2 : 2 3 5 6
Enter Max array for P3 : 0 6 5 2
Enter Max array for P4 : 0 6 5 6

Enter Available array
1 5 2 0

        Need Matrix
        R0      R1      R2      R3
P0      0       0       0       0
P1      0       7       5       0
P2      1       0       0       2
P3      0       0       2       0
P4      0       6       4       2

Do you want to check safety? (1-yes 2-no)
your choice : 1
Yes , Safe state <P0,P2,P3,P4,P1>

Do you want to make request? (1-yes 2-no)
your choice : 1
enter index of process: 1
enter request of process 1 : 0 4 2 0

        Need Matrix
        R0      R1      R2      R3
P0      0       0       0       0
P1      0       3       3       0
P2      1       0       0       2
P3      0       0       2       0
P4      0       6       4       2

Yes , Safe state <P1req,P0,P2,P3,P4,P1>

Do you want to test agian?(y/n)
```

GUI:

Check the safety of the system.

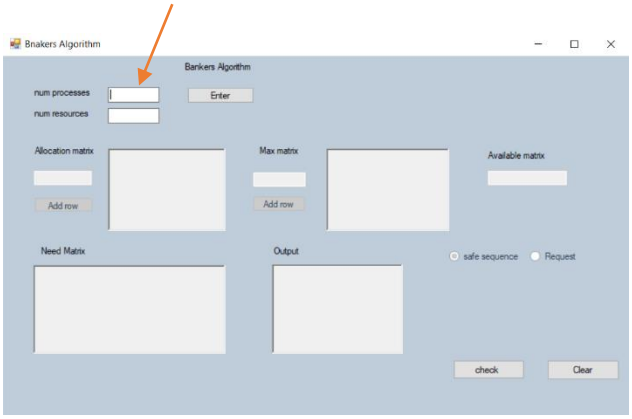


Check the request.

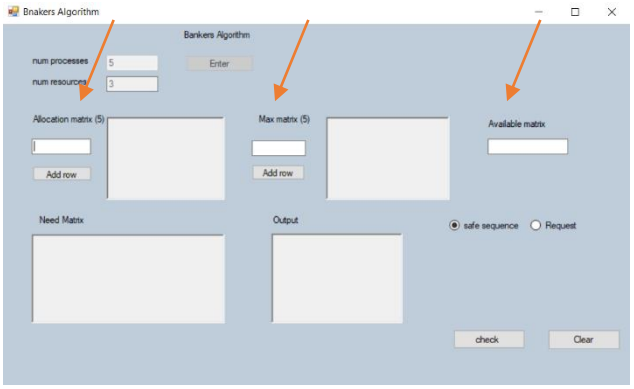


How to use the GUI:

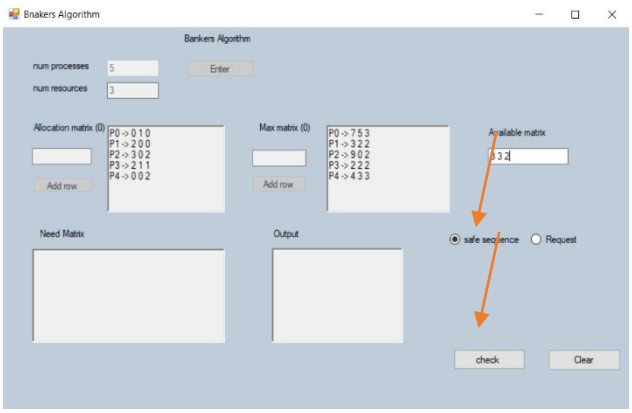
- First Enter the number of processes and resources in their fields
- Then, press Enter.

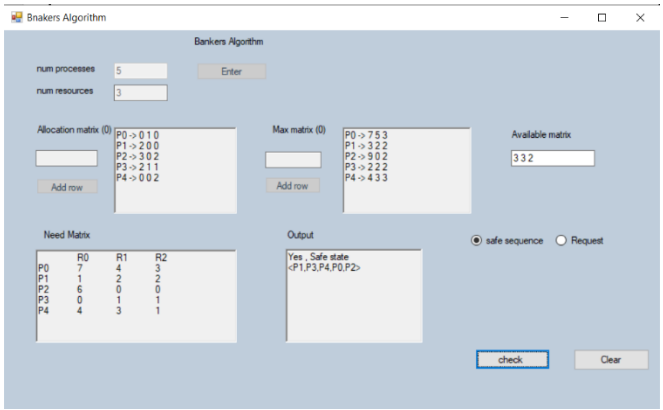


- Now the Allocation matrix, Max matrix and Available matrix will be Enabled.
- In the Allocation matrix and Max matrix you will need to press (Add row) button after entering each row in the test field. (enter values with space separated ex: 1 2 3).
- After that enter the Available matrix space separated values.

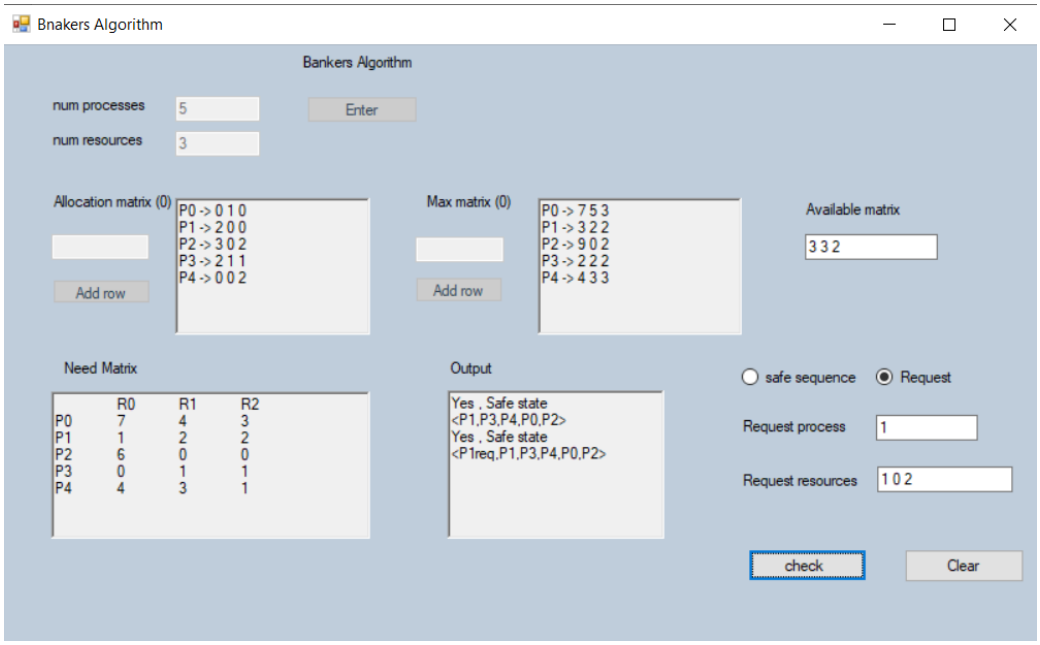


- Now if you want to check the safety, you need to check the safety option and click on check button.





- If you want to check request, you need to choose the request option and insert your request in the specified fields. Then, press check button.



- At the end if you want to check another system press clear button to restart it.

Link of files:

https://github.com/ahmed192a/bankers-Algorithm.git

Test cases used:

```
3
5
0 1 0
2 0 0
3 0 2
2 1 1
0 0 2
7 5 3
3 2 2
9 0 2
2 2 2
4 3 3
3 3 2
```

```
3
5
0 4 0
2 0 0
3 0 2
2 1 1
0 0 2
7 5 3
3 2 2
9 0 2
2 2 2
4 3 3
3 0 2
```

```
4
5
0 0 1 2
1 0 0 0
1 3 5 4
0 6 3 2
0 0 1 4
0 0 1 2
1 7 5 0
2 3 5 6
0 6 5 2
0 6 5 6
1 5 2 0
```