BANKER'S ALGORITHM

Dr.Soha Eng.Nancy

Mohamed Taha Salah Hassan 192000280

Contents

Introduction 2
Banker's Algorithm Definition 2
Why Banker's algorithm is named so? 2
Code contents 2
Code Cases 2
The contents of the input file 3
(Read from the file) output: 3
(User define) output: 4

Table of figures

Figure 1 2
Figure 2 3
Figure 3 3
Figure 4 4

Introduction

Banker's Algorithm Definition

The banker's algorithm is a resource allocation and deadlock avoidance algorithm that tests for safety by simulating the allocation for predetermined maximum possible amounts of all resources, then makes an "s-state" check to test for possible activities, before deciding whether allocation should be allowed to continue.

Why Banker's algorithm is named so?

Banker's algorithm is named so because it is used in banking system to check whether loan can be sanctioned to a person or not. Suppose there are n number of account holders in a bank and the total sum of their money is S. If a person applies for a loan, then the bank first subtracts the loan amount from the total money that bank has and if the remaining amount is greater than S then only the loan is sanctioned. It is done because if all the account holders come to withdraw their money, then the bank can easily do it.

In other words, the bank would never allocate its money in such a way that it can no longer satisfy the needs of all its customers. The bank would try to be in safe state always.

Code contents

The code contains three functions:

1. `void Read_from_file();`
2. `void User_define();`
3. `void main();`

Code Cases

Figure 1

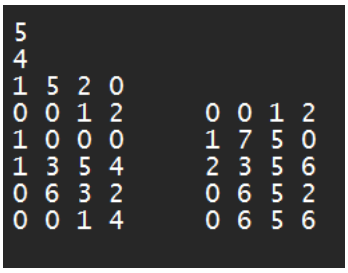
```
1.Read from file                2.User define
please enter your choice: _
```

This code can take its input in two ways:

1. The first is via a pre-configured file, and I will explain the contents of the file in this report.
2. The other way is by taking input from the user.

The contents of the input file

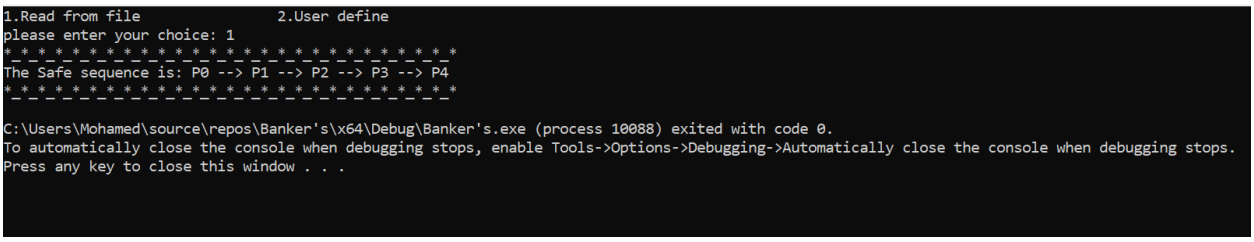
Figure 2



1. The first row contains the number of processes.
2. The second row contains the number of resources.
3. The third row contains the available matrix.
4. From the fourth row to the end of the file, it contains two arrays, the first is the allocation matrix, and the second is the max array.

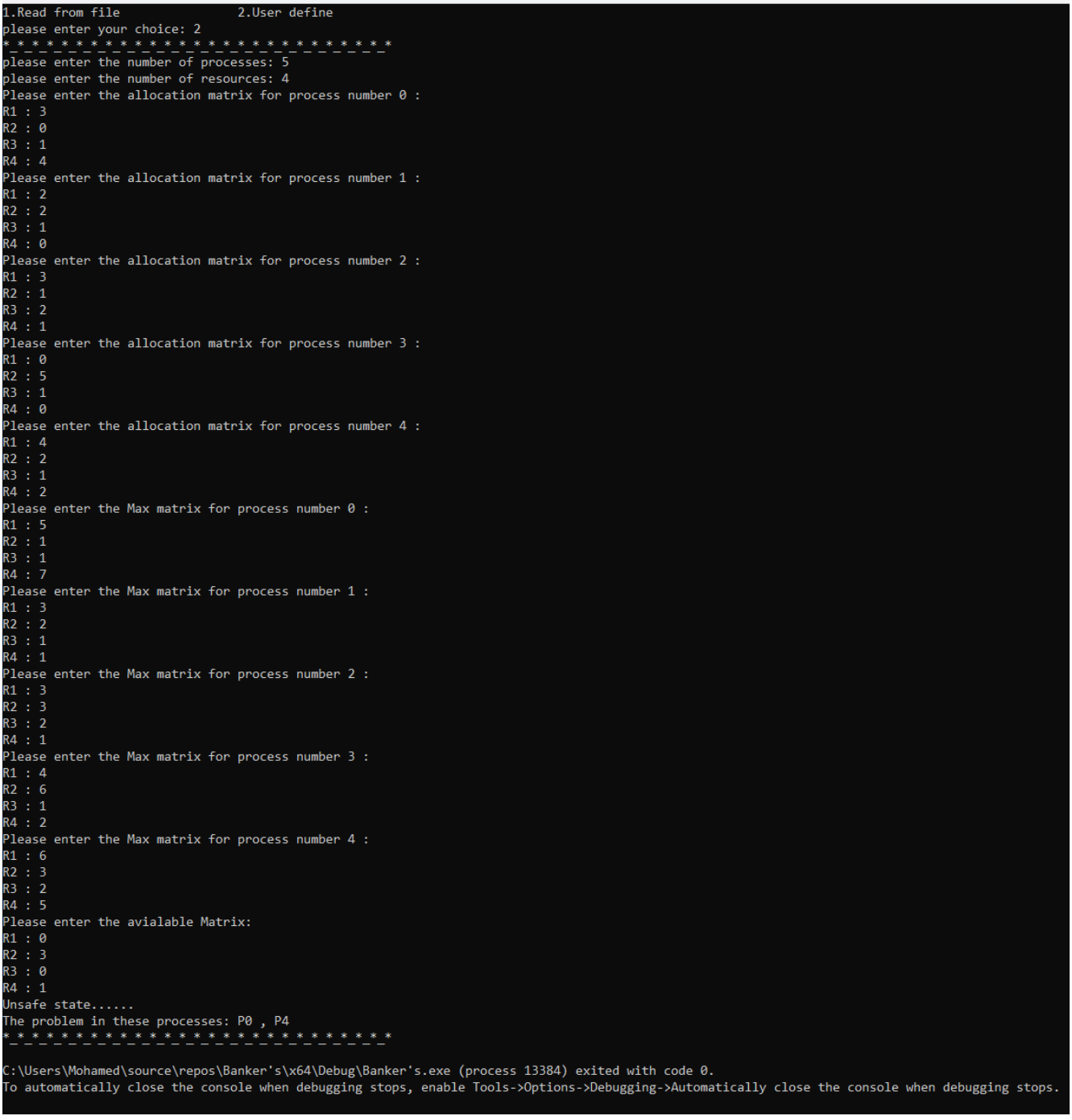
(Read from the file) output:

Figure 3



(User define) output:

Figure 4



```
1.Read from file               2.User define
please enter your choice: 2
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
please enter the number of processes: 5
please enter the number of resources: 4
Please enter the allocation matrix for process number 0 :
R1 : 3
R2 : 0
R3 : 1
R4 : 4
Please enter the allocation matrix for process number 1 :
R1 : 2
R2 : 2
R3 : 1
R4 : 0
Please enter the allocation matrix for process number 2 :
R1 : 3
R2 : 1
R3 : 2
R4 : 1
Please enter the allocation matrix for process number 3 :
R1 : 0
R2 : 5
R3 : 1
R4 : 0
Please enter the allocation matrix for process number 4 :
R1 : 4
R2 : 2
R3 : 1
R4 : 2
Please enter the Max matrix for process number 0 :
R1 : 5
R2 : 1
R3 : 1
R4 : 7
Please enter the Max matrix for process number 1 :
R1 : 3
R2 : 2
R3 : 1
R4 : 1
Please enter the Max matrix for process number 2 :
R1 : 3
R2 : 3
R3 : 2
R4 : 1
Please enter the Max matrix for process number 3 :
R1 : 4
R2 : 6
R3 : 1
R4 : 2
Please enter the Max matrix for process number 4 :
R1 : 6
R2 : 3
R3 : 2
R4 : 5
Please enter the avialable Matrix:
R1 : 0
R2 : 3
R3 : 0
R4 : 1
Unsafe state......
The problem in these processes: P0 , P4
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

C:\Users\Mohamed\source\repos\Banker's\x64\Debug\Banker's.exe (process 13384) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
```