**Lovely Professional University**

**Operating System Assignment Simulation**

**Submitted To: Keshav Dhir (25128)**

**Submitted By:**

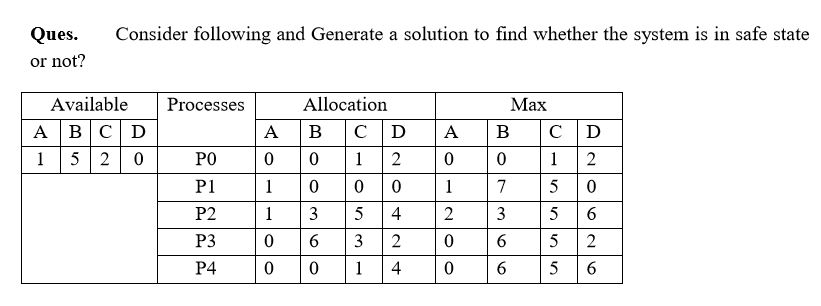
**Student Name : Sneha Singh**

**Student ID : 11808742**

**Email Address:** [**sneha.11808742@lpu.in**](mailto:sneha.11808742@lpu.in)

**GitHub Link : ssneha769**

**Code : Banker’s Algorithm**

**Problem** **:**

**Problem Description:**

In above problem I have to solve the problem to check whether the system is safe or not.

To solve this problem I have to use Banker’s Algorithm. In this firstly I made one matrix named **Need** (Need=Max-Allocation) .After solving this problem using banker’s algorithm the system achieve deadlock situation which shows that the system is

not in safe state.the sequence of process is{P0,P2,P3} after that the system achieve deadlock situation.

**Banker’s Algorithm:**

The banker’s algorithm  which is also known as avoidance algorithm is a deadlock detection algorithm. It was developed by Edsger Dijkstra. It is designed to check the safe state whenever a resource is requested. It takes analogy of bank, where customer request to withdraw cash. Based on some data the cash is lent to the customer. The banker can’t give more cash than what the customer has requested for, and the total available cash.  As this algorithm uses bank analogy so named as banker’s algorithm.

**Description of Banker’s Algorithm:**

1) Let Work and Finish be vectors of length ‘m’ and ‘n’ respectively.  
Initialize: Work = Available  
Finish[i] = false; for i=1, 2, 3, 4….n

2) Find an i such that both  
a) Finish[i] = false  
b) Needi <= Work  
if no such i exists goto step (4)

3) Work = Work + Allocation[i]  
Finish[i] = true  
goto step (2)

4) if Finish [i] = true for all i  
then the system is in a safe state

**Complexcity :**r\*(p\*p) [p=no. of active process,r=no. of resource]

**Constraints used:**

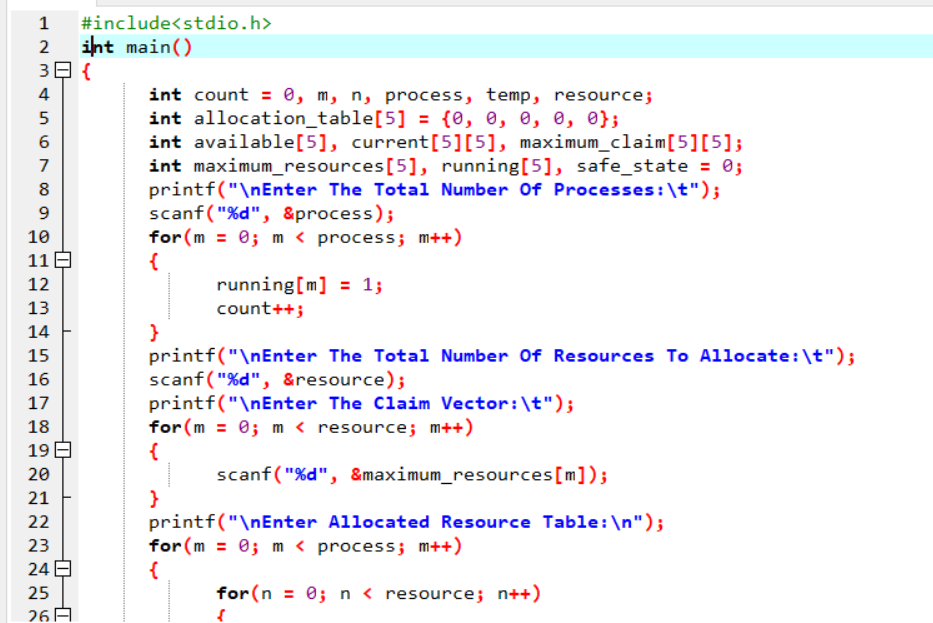
**Allocation** : indicates where process you have received a resource.

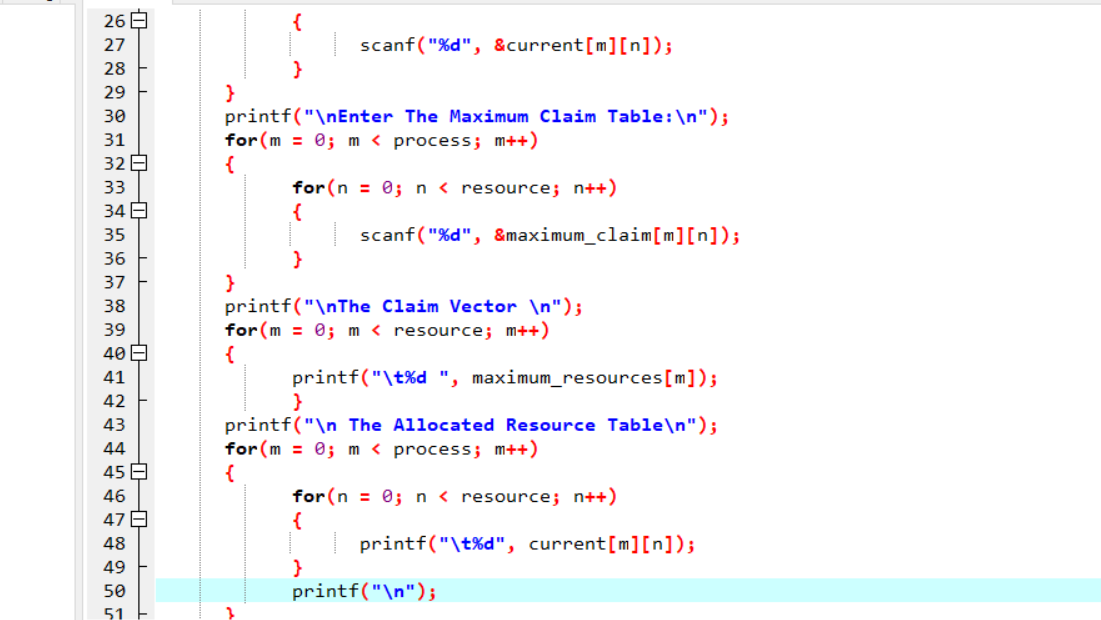
**Need** : Express how many more resources can be allocated in future.

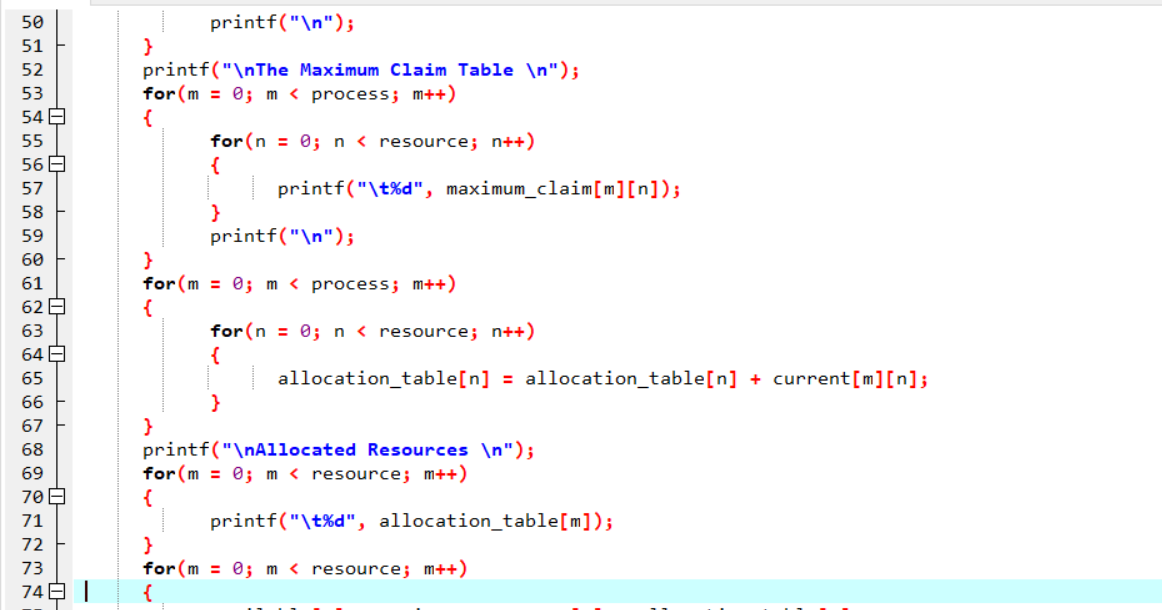
**Available :** Indicates which resource is available.

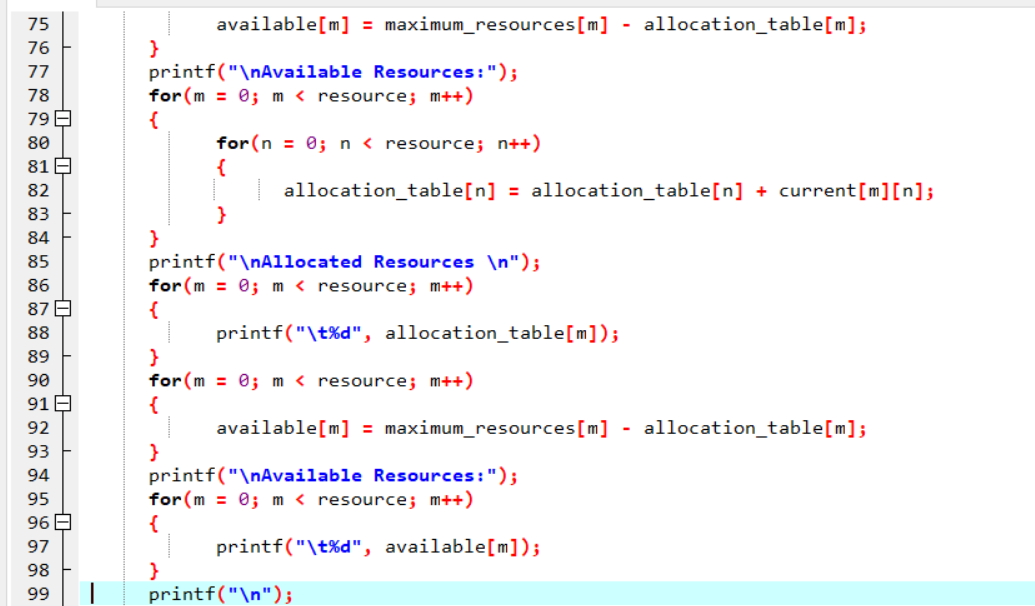
**Max** :Expression of the maximum number of resources.

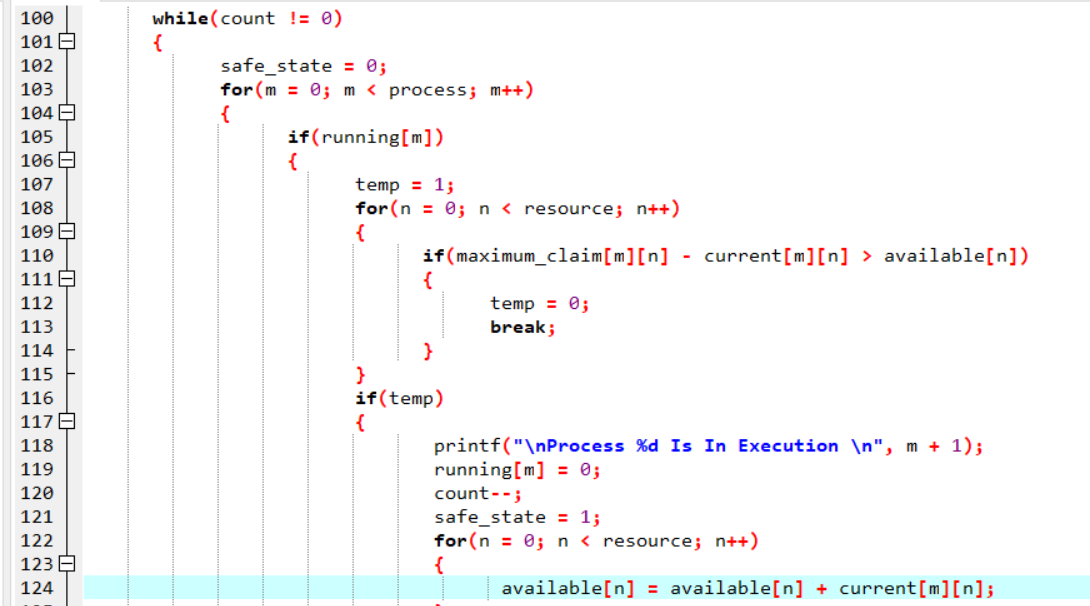
**CODE SNIPET:**

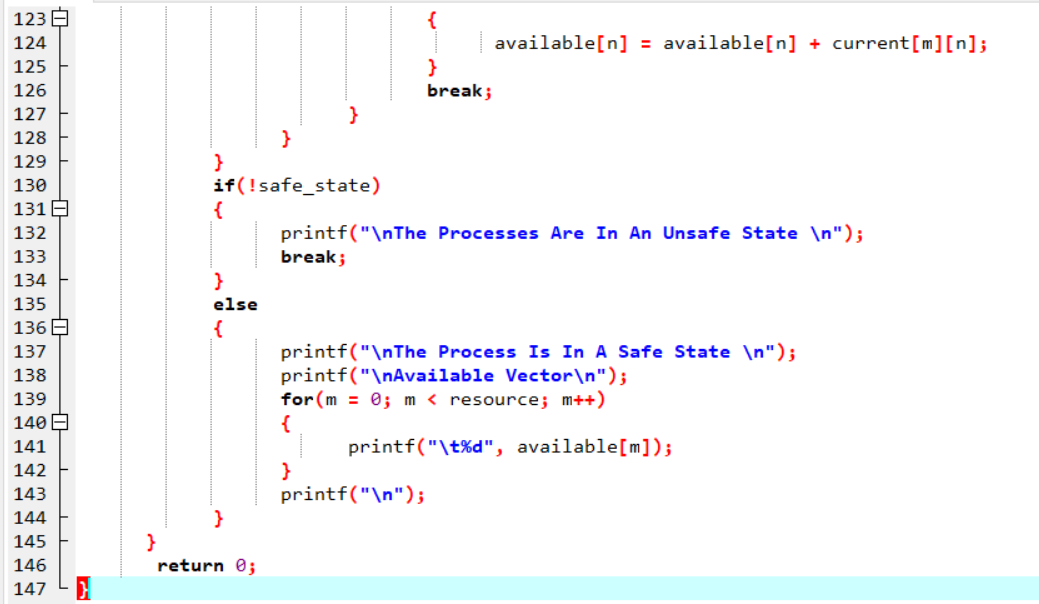




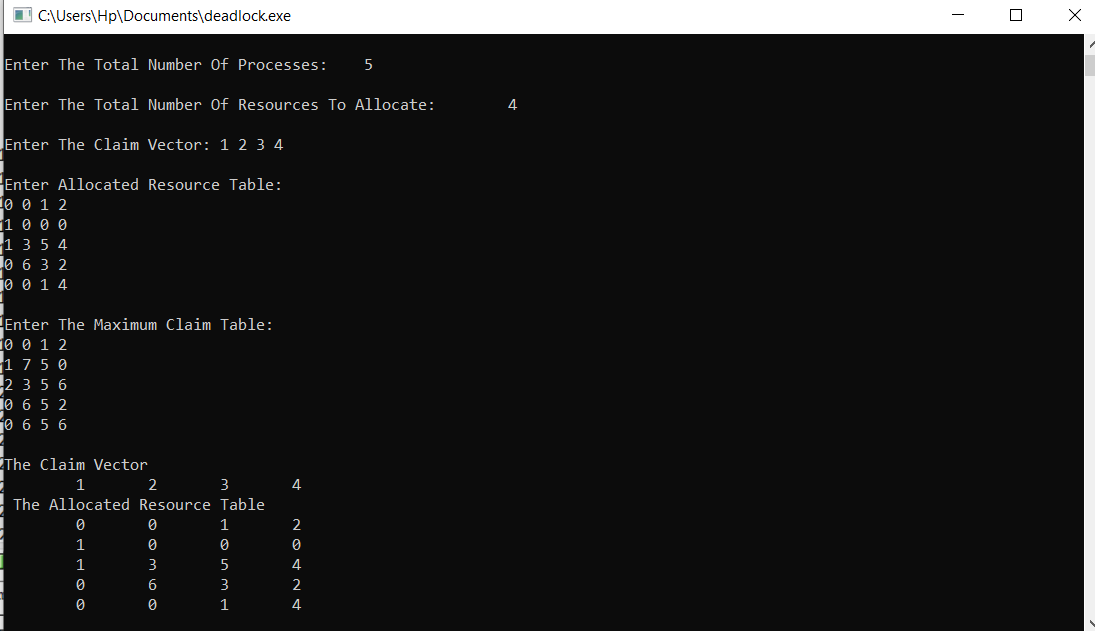


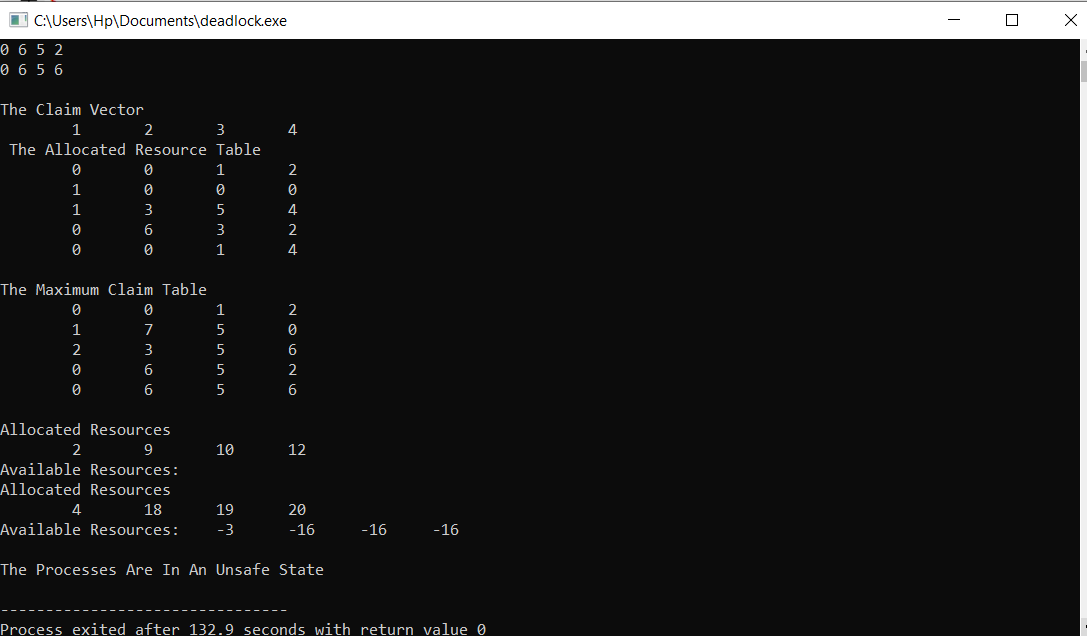






**Output:**





**TEST CASES:**

