Assignment 5

Code

```
/*
Problem Statement - Implement C program for Deadlock Avoidance: Banker’sAlgorithm
*/

#include <bits/stdc++.h>
using namespace std;
#define NO_OF_RESOURCE_INSTANCES 3
#define NO_OF_PROCESSES 5

struct process
{
    int rank = 0, allocation[NO_OF_RESOURCE_INSTANCES],
max_need[NO_OF_RESOURCE_INSTANCES], need[NO_OF_RESOURCE_INSTANCES] = { 0 };
};

int available[NO_OF_RESOURCE_INSTANCES] = { 2, 1, 0 };
process given[NO_OF_PROCESSES] = {
    {0, {1, 1, 2}, {4, 3, 3}, {0}},
    {0, {2, 1, 2}, {3, 2, 2}, {0}},
    {0, {4, 0, 1}, {9, 0, 2}, {0}},
    {0, {0, 2, 0}, {7, 5, 3}, {0}},
    {0, {1, 1, 2}, {1, 1, 2}, {0}} };

void display_resource_instances(int instances[], int no_of_instances)
{
    for (int i = 0; i < no_of_instances; i++)
    {
        cout << instances[i] << " ";
    }
    return;
}

void display_table(process given[])
{
    cout << "\n------------------------------------------------";
    cout << "\n| Processes | Allocation |   Max   | Available | ";
    cout << "\n------------------------------------------------";
    // cout << "\n|    P1     |   4 3 3    | 4 3 3 |   2 1 0   |";
    int no_of_completed_processes = 1;
    while (no_of_completed_processes <= NO_OF_PROCESSES)
    {
        bool break_while = true;
        size_t process;
        for (process = 0; process < NO_OF_PROCESSES; process++)
        {
            if (given[process].rank == no_of_completed_processes)
            {
                cout << "\n|    P" << process + 1 << "     |   ";
                display_resource_instances(given[process].allocation,
NO_OF_RESOURCE_INSTANCES);
                cout << "   |  ";
                display_resource_instances(given[process].max_need,
NO_OF_RESOURCE_INSTANCES);
                cout << " |   ";
                display_resource_instances(available, NO_OF_RESOURCE_INSTANCES);
                cout << "  |";
                cout << "\n------------------------------------------------";
                no_of_completed_processes++;
                break_while = false;
            }
        }
        if (break_while && (process == NO_OF_PROCESSES))
            break;
    }
    cout << endl;
};

int main()
{
    for (int i = 0; i < NO_OF_PROCESSES; i++)
    {
        for (int j = 0; j < NO_OF_RESOURCE_INSTANCES; j++)
        {
            given[i].need[j] = given[i].max_need[j] - given[i].allocation[j];
        }
    }
    /**
    need[5][3] = {
       3 2 1,
       1 1 0,
       5 0 1,
       7 3 3,
       0 0 0,
    }
    */

    int isSafeState, isStarving, rank = 0;
    // cout << "Process sequence: ";
    while (true)
    {
        isSafeState = 0;
        bool check_process;
        for (int process = 0; process < NO_OF_PROCESSES; process++)
        {
            if (given[process].rank == 0)
            {
                check_process = true;
                for (int resource = 0; resource < NO_OF_RESOURCE_INSTANCES;
resource++)
                {
                    if (given[process].need[resource] > available[resource])
                    {
                        check_process = false;
                        break;
                    }
                }
                if (!check_process)
                    continue;
                for (int resource = 0; resource < NO_OF_RESOURCE_INSTANCES;
resource++)
                {
                    available[resource] += given[process].allocation[resource];
                    given[process].allocation[resource] +=
given[process].need[resource];
                    given[process].need[resource] = 0;
                }
                given[process].rank = ++rank;
                display_table(given);
                isSafeState = 1;
                // cout << "P" << process + 1 << " ";
            }
        }
        isStarving = 0;
        for (int process = 0; process < NO_OF_PROCESSES; process++)
        {
            if (given[process].rank != 0)
            {
                isStarving++;
            }
            else if (!isSafeState)
            {
                cout << "Deadlock condition!" << endl;
                return 0;
            }
        }
        if (isStarving >= NO_OF_PROCESSES)
        {
            display_table(given);
            cout << "\nAll processes finished, CPU in idle state..." << endl;
            return 0;
        }
    }
};
```

Output

```
abhishek-jadhav@abhishek-jadhav-ubuntu:~/Codes/OS Assignments/33232$ ./a.out

------------------------------------------------
| Processes | Allocation |   Max    | Available |
------------------------------------------------
|    P2     |   3 2 2    |  3 2 2  |   4 2 2   |
------------------------------------------------

------------------------------------------------
| Processes | Allocation |   Max    | Available |
------------------------------------------------
|    P2     |   3 2 2    |  3 2 2  |   5 3 4   |
------------------------------------------------
|    P5     |   1 1 2    |  1 1 2  |   5 3 4   |
------------------------------------------------

------------------------------------------------
| Processes | Allocation |   Max    | Available |
------------------------------------------------
|    P2     |   3 2 2    |  3 2 2  |   6 4 6   |
------------------------------------------------
|    P5     |   1 1 2    |  1 1 2  |   6 4 6   |
------------------------------------------------
|    P1     |   4 3 3    |  4 3 3  |   6 4 6   |
------------------------------------------------

------------------------------------------------
| Processes | Allocation |   Max    | Available |
------------------------------------------------
|    P2     |   3 2 2    |  3 2 2  |   10 4 7   |
------------------------------------------------
|    P5     |   1 1 2    |  1 1 2  |   10 4 7   |
------------------------------------------------
|    P1     |   4 3 3    |  4 3 3  |   10 4 7   |
------------------------------------------------
```

```
|    P3    |   9 0 2    | 9 0 2 |   10 4 7   |
------------------------------------------------

------------------------------------------------
| Processes | Allocation |   Max   | Available |
------------------------------------------------
|    P2    |   3 2 2    | 3 2 2 |   10 6 7   |
------------------------------------------------
|    P5    |   1 1 2    | 1 1 2 |   10 6 7   |
------------------------------------------------
|    P1    |   4 3 3    | 4 3 3 |   10 6 7   |
------------------------------------------------
|    P3    |   9 0 2    | 9 0 2 |   10 6 7   |
------------------------------------------------
|    P4    |   7 5 3    | 7 5 3 |   10 6 7   |
------------------------------------------------

------------------------------------------------
| Processes | Allocation |   Max   | Available |
------------------------------------------------
|    P2    |   3 2 2    | 3 2 2 |   10 6 7   |
------------------------------------------------
|    P5    |   1 1 2    | 1 1 2 |   10 6 7   |
------------------------------------------------
|    P1    |   4 3 3    | 4 3 3 |   10 6 7   |
------------------------------------------------
|    P3    |   9 0 2    | 9 0 2 |   10 6 7   |
------------------------------------------------
|    P4    |   7 5 3    | 7 5 3 |   10 6 7   |
------------------------------------------------

All processes finished, CPU in idle state...
abhishek-jadhav@abhishek-jadhav-ubuntu:~/Codes/OS Assignments/33232$
```