Assignment No. 5

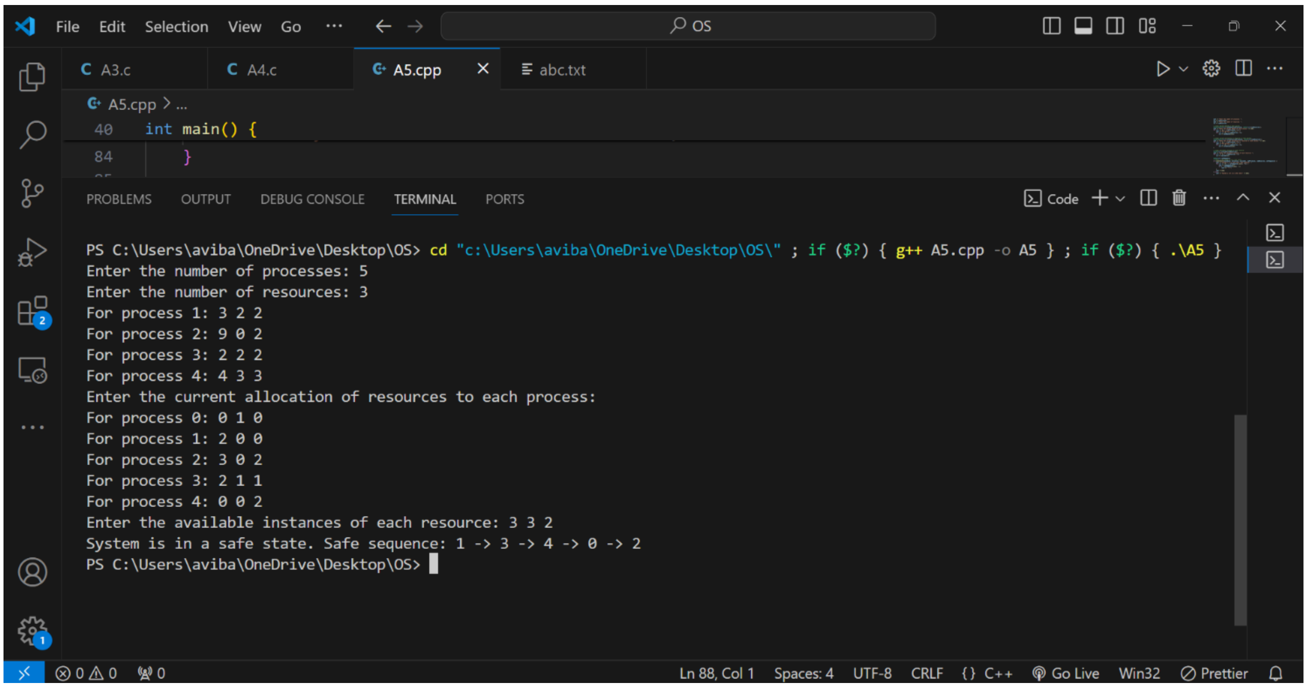
Write a program to implement Banker's Algorithm for deadlock avoidance.

Code:

```
#include <iostream>

#include <vector>


using namespace std;

bool isSafeState(const vector<vector<int>>& maxDemand, const vector<vector<int>>& allocation,
const vector<int>& available, int numProcesses, int numResources, vector<int>& safeSequence) {

  vector<int> work = available;

  vector<bool> finish(numProcesses, false);


  int count = 0;

  while (count < numProcesses) {

    bool found = false;

    for (int i = 0; i < numProcesses; ++i) {

      if (!finish[i]) {

        bool canAllocate = true;

        for (int j = 0; j < numResources; ++j) {

          if (maxDemand[i][j] - allocation[i][j] > work[j]) {

            canAllocate = false;

            break;

          }

        }

        if (canAllocate) {

          for (int j = 0; j < numResources; ++j)

            work[j] += allocation[i][j];
```

```
                safeSequence.push_back(i);
                finish[i] = true;
                ++count;
                found = true;
            }
        }
    }
    if (!found)
        break;
}

    return count == numProcesses;
}

int main() {
    int numProcesses, numResources;

    cout << "Enter the number of processes: ";
    cin >> numProcesses;
    cout << "Enter the number of resources: ";
    cin >> numResources;

    vector<vector<int>> maxDemand(numProcesses, vector<int>(numResources));
    cout << "Enter the maximum demand of each process:" << endl;
    for (int i = 0; i < numProcesses; ++i) {
        cout << "For process " << i << ": ";
        for (int j = 0; j < numResources; ++j)
            cin >> maxDemand[i][j];
    }
    vector<vector<int>> allocation(numProcesses, vector<int>(numResources));
    cout << "Enter the current allocation of resources to each process:" << endl;
    for (int i = 0; i < numProcesses; ++i) {
        cout << "For process " << i << ": ";
        for (int j = 0; j < numResources; ++j)
            cin >> allocation[i][j];
    }
    vector<int> available(numResources);
    cout << "Enter the available instances of each resource: ";
    for (int i = 0; i < numResources; ++i)
        cin >> available[i];

    vector<int> safeSequence;
    if (isSafeState(maxDemand, allocation, available, numProcesses, numResources, safeSequence)) {
        cout << "System is in a safe state. Safe sequence: ";
        for (int i = 0; i < safeSequence.size(); ++i) {
            cout << safeSequence[i];
            if (i < safeSequence.size() - 1)
                cout << " -> ";
        }
        cout << endl;
    } else {
        cout << "System is not in a safe state." << endl;
    }

    return 0;
}
```

Output :


```
PS C:\Users\aviba\OneDrive\Desktop\OS> cd "c:\Users\aviba\OneDrive\Desktop\OS\" ; if ($?) { g++ A5.cpp -o A5 } ; if ($?) { .\A5 }
Enter the number of processes: 5
Enter the number of resources: 3
For process 1: 3 2 2
For process 2: 9 0 2
For process 3: 2 2 2
For process 4: 4 3 3
Enter the current allocation of resources to each process:
For process 0: 0 1 0
For process 1: 2 0 0
For process 2: 3 0 2
For process 3: 2 1 1
For process 4: 0 0 2
Enter the available instances of each resource: 3 3 2
System is in a safe state. Safe sequence: 1 -> 3 -> 4 -> 0 -> 2
PS C:\Users\aviba\OneDrive\Desktop\OS>
```
File Edit Selection View Go
OS
A3.c
A4.c
A5.cpp
abc.txt
A5.cpp > ...
int main() {
}
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Code
Ln 88, Col 1 Spaces: 4 UTF-8 CRLF {} C++ Go Live Win32 Prettier