```
#include <iostream>
using namespace std;

int main()
{
    int n, m, i, j, k;
    cout<<"no of processes=";
    cin>>n;
    cout<<"no of resources=";
    cin>>m;
    int alloc[n][m];
    cout<<"enter allocation matrix=";
    for(int i=0;i<n;i++){
        for(int j=0;j<m;j++){
            cin>>alloc[i][j];
        }
    }
    int max[n][m];
    cout<<"enter max matrix=";
    for(int i=0;i<n;i++){
        for(int j=0;j<m;j++){
            cin>>max[i][j];
        }
    }

    int avail[m];
    for(int i=0;i<m;i++){
        cout<<"enter"<< i <<"element of the availability=";
        cin>>avail[i];
    }

    cout<<"availability matrix=\n";
    int f[n], ans[n], ind = 0;
    for (k = 0; k < n; k++) {
        f[k] = 0;
    }
    int need[n][m];
    for (i = 0; i < n; i++) {
        for (j = 0; j < m; j++)
            need[i][j] = max[i][j] - alloc[i][j];
    }
    int y = 0;
    for (k = 0; k < n; k++) {
        for (i = 0; i < n; i++) {
```

```
            if (f[i] == 0) {
                int flag = 0;
                for (j = 0; j < m; j++) {
                    if (need[i][j] > avail[j]){
                        flag = 1;
                    break;
                    }
                }

                if (flag == 0) {
                    ans[ind++] = i;
                    for (y = 0; y < m; y++){
                        avail[y] += alloc[i][y];
                        cout<<"\t";
                        cout<<avail[y];
                    }
                    cout<<"\n";
                    f[i] = 1;
                }
            }
    }
}
    cout << "process execution order=" << endl;
    for (i = 0; i < n-1 ; i++)
        cout << " P" << ans[i] << " ->";
        cout<<"p"<<ans[n-1];
    return (0);
}

#include <iostream>
using namespace std;

int main()
{
    int n, m, i, j, k;
    cout<<"no of processes=";
    cin>>n;
    cout<<"no of resources=";
    cin>>m;
    int alloc[n][m];
    cout<<"enter allocation matrix=";
    for(int i=0;i<n;i++){
        for(int j=0;j<m;j++){
            cin>>alloc[i][j];
        }
    }
    int max[n][m];
    cout<<"enter max matrix=";
    for(int i=0;i<n;i++){
        for(int j=0;j<m;j++){
            cin>>max[i][j];
```

```
        }
    }

    int avail[m];
    for(int i=0;i<m;i++){
        cout<<"enter"<< i <<"element of the availability=";
        cin>>avail[i];
    }

    cout<<"availability matrix=\n";
    int f[n], ans[n], ind = 0;
    for (k = 0; k < n; k++) {
        f[k] = 0;
    }
    int need[n][m];
    for (i = 0; i < n; i++) {
        for (j = 0; j < m; j++)
            need[i][j] = max[i][j] - alloc[i][j];
    }
    int y = 0;
    for (k = 0; k < n; k++) {
        for (i = 0; i < n; i++) {
            if (f[i] == 0) {
                int flag = 0;
                for (j = 0; j < m; j++) {
                    if (need[i][j] > avail[j]){
                        flag = 1;
                    break;
                    }
                }

                if (flag == 0) {
                    ans[ind++] = i;
                    for (y = 0; y < m; y++){
                        avail[y] += alloc[i][y];
                        cout<<"\t";
                        cout<<avail[y];
                    }
                    cout<<"\n";
                    f[i] = 1;
                }
            }
    }
}
    cout << "process execution order=" << endl;
    for (i = 0; i < n-1 ; i++)
        cout << " P" << ans[i] << " ->";
        cout<<"p"<<ans[n-1];
    return (0);
}
```

OUTPUT:-

```
 banker } ; if ($?) { .\banker }
no of processes=5
no of resources=3
enter allocation matrix=0 1 0
2 0 0
3 0 2
2 1 1
0 0 2
enter max matrix=7 5 3
3 2 2
9 0 2
4 2 2
5 3 3
enter0element of the availability=3
enter1element of the availability=3
enter2element of the availability=2
availability matrix=
        5       3       2
        7       4       3
        7       4       5
        7       5       5
        10      5       7
process execution order=
 P1 -> P3 -> P4 -> P0 ->p2
PS C:\Users\AJAY SHARMA\Downloads>
```