**Banker's Algorithm**

#include <iostream>

using namespace std;

int main()

{

int n, m, i, j, k;

cout<<"enter the processes";

cin>>n;

cout<<"enter the resources";

cin>>m;

int alloc[n][m];

cout<<"enter the allocaton matrix";

for(i=0;i<n;i++)

for(j=0;j<m;j++)

cin>>alloc[i][j];

int max[n][m];

cout<<"enter the maximum matrix";

for(i=0;i<n;i++)

for(j=0;j<m;j++)

cin>>max[i][j];

int avail[m];

cout<<"enter the Available Resources";

for(j=0;j<m;j++)

cin>>avail[j];

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 < 5; 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];

f[i] = 1;

}

}

}

}

cout << "Following is the SAFE Sequence" << endl;

for (i = 0; i < n - 1; i++)

cout << " P" << ans[i] << " ->";

cout << " P" << ans[n - 1] <<endl;

return (0);

}

