Bankers algorithm

#include <stdio.h>

int main()

{

int n, m, i, j, k;

printf("enter the processes:");

scanf("%d",&n);

printf("enter the resources:");

scanf("%d",&m);

int alloc[4][3] = {{1,0,0},{6,1,2},{2,1,1},{0,0,2}};

int claim[4][3] = {{3,2,2},{6,1,3},{3,1,4},{4,2,2}};

int resourcevector[3] = {9,3,6};

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] = claim[i][j] - alloc[i][j];

}

int y = 0;

for (k = 0; k < 4; k++)

{

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

{

if (f[i] == 0)

{

int flag = 0;

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

{

if (need[i][j] > resourcevector[j])

{

flag = 1;

break;

}

}

if (flag == 0)

{

ans[ind++] = i;

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

resourcevector[y] += alloc[i][y];

f[i] = 1;

}

}

}

}

int flag = 1;

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

{

if(f[i]==0)

{

flag=0;

printf("The following system is not safe");

break;

}

}

if(flag==1)

{

printf("Following is the SAFE Sequence\n");

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

printf(" P%d ->", ans[i]);

printf(" P%d", ans[n - 1]);

}

return 0;

}

Output

