#include <stdio.h>

int n, m;

void display(int mat[5][3])

{

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

{

printf("\n");

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

{

printf("%d ", mat[i][j]);

}

}

}

int main()

{

printf("\n\n Enter number of process: ");

scanf("%d", &n);

printf("\n\n Enter number of resources: ");

scanf("%d", &m);

int alloc[5][3], need[5][3], max[5][3];

int available[3];

printf("\n\n Enter Allocation Matrix:\n");

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

{

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

{

scanf("%d", &alloc[i][j]);

}

}

printf("\n\n Enter Max Matrix:\n");

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

{

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

{

scanf("%d", &max[i][j]);

}

}

printf("\n\n Enter available Array:\n");

for (int i = 0; i < m; i++)

{

scanf("%d", &available[i]);

}

printf("\n\n Displaying Allocation Matrix:");

display(alloc);

printf("\n\n Displaying Max Matrix:");

display(max);

printf("\n\n Displaying Available Array:\n");

for (int i = 0; i < m; i++)

{

printf("%d ", available[i]);

}

// Step 1: Creating Need matrix

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

{

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

{

need[i][j] = max[i][j] - alloc[i][j];

}

}

printf("\n\n Displaying Need Matrix:");

display(need);

// Step 2: Creating Flag variable to keep track of which process complete

int flag[5];

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

{

flag[i] = 0;

}

// Step 3: Allocating Resources if available is greater than need

int safe\_seq[5], ct = 0;

int i, j, k, f;

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

{

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

{

f = 1;

if (flag[i] == 1)

{

continue;

}

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

{

if (available[j] < need[i][j])

{

f = 0;

break;

}

}

if (f == 1)

{

flag[i] = 1;

break;

}

}

if (i == n)

{

continue;

}

if (flag[i] == 1)

{

safe\_seq[ct] = i;

ct++;

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

{

available[j] += alloc[i][j];

}

}

}

f = 0;

// Checking if Safe Sequence is present or not

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

{

if (flag[i] == 0)

{

printf("\n\n Safe Sequence Not Present!");

f = 1;

break;

}

}

// Printing Safe Sequence

if (f == 0)

{

printf("\n\n Safe Sequence: ");

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

{

printf("P%d, ", safe\_seq[i] + 1);

}

}

return 0;

}



