#include <stdio.h>

#include <stdlib.h>

#include <stdbool.h>

// Function to print the matrix

void printMatrix(int rows, int cols, int matrix[rows][cols]) {

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

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

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

}

printf("\n");

}

}

// Banker's Algorithm to find the safe sequence

bool BankerAlg(int n, int m, int Allocation[n][m], int Available[m], int Max[n][m], int Need[n][m], int flag[n], int answer[n]) {

int a\_i = 0;

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

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

if (flag[i] == 0) {

int temp = 0;

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

if (Need[i][j] > Available[j]) {

temp = 1;

break;

}

}

if (temp == 0) {

answer[a\_i++] = i;

flag[i] = 1;

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

Available[j] += Allocation[i][j];

}

break;

}

}

}

}

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

if (flag[i] == 0) {

return false;

}

}

return true;

}

int main() {

int n, m;

printf("Please enter the number of processes: ");

scanf("%d", &n);

printf("Please enter the number of resources: ");

scanf("%d", &m);

int Available[m];

int flag[n];

int Max[n][m];

int Allocation[n][m];

int Need[n][m];

int answer[n];

// Initialize the flag and answer arrays

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

flag[i] = 0;

answer[i] = -1;

}

// Randomly initialize Available, Max, and Allocation matrices

srand(time(NULL));

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

Available[i] = rand() % 5 + 1;

}

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

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

Max[i][j] = rand() % 10 + 1;

Allocation[i][j] = rand() % (Max[i][j] + 1);

Need[i][j] = Max[i][j] - Allocation[i][j];

}

}

// Print matrices

printf("The Available array: ");

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

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

}

printf("\n\n");

printf("Max Matrix:\n");

printMatrix(n, m, Max);

printf("Allocation Matrix:\n");

printMatrix(n, m, Allocation);

printf("Need Matrix:\n");

printMatrix(n, m, Need);

printf("Finding Safe sequence using Banker's Algorithm...\n");

if (BankerAlg(n, m, Allocation, Available, Max, Need, flag, answer)) {

printf("The Safe sequence is:\n");

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

if (i == n - 1) {

printf("Process[%d]\n", answer[i]);

} else {

printf("Process[%d]->", answer[i]);

}

}

} else {

printf("Safe Sequence Doesn't Exist...\n");

}

return 0;

}