```c
#include <stdio.h>

#define MAX_PROCESSES 10
#define MAX_RESOURCES 10

// Function to find safe sequence using Banker's algorithm
int findSafeSequence(int n, int m, int max[MAX_PROCESSES][MAX_RESOURCES], int allocation[MAX_PROCESSES][MAX_RESOURCES], int available
[MAX_RESOURCES], int safeSeq[]) {
    int need[MAX_PROCESSES][MAX_RESOURCES];
    int finish[MAX_PROCESSES] = {0};
    int work[MAX_RESOURCES];
    int i, j, k;

    // Calculate the need matrix
    for (i = 0; i < n; ++i) {
        for (j = 0; j < m; ++j) {
            need[i][j] = max[i][j] - allocation[i][j];
        }
    }

    // Initialize work array
    for (i = 0; i < m; ++i) {
        work[i] = available[i];
    }

    int count = 0;
    while (count < n) {
        int found = 0;
        for (i = 0; i < n; ++i) {
            if (!finish[i]) {
                int flag = 1;
                for (j = 0; j < m; ++j) {
                    if (need[i][j] > work[j]) {
                        flag = 0;
                        break;
                    }
                }
                if (flag) {
                    for (k = 0; k < m; ++k) {
                        work[k] += allocation[i][k];
                    }
                    safeSeq[count++] = i;
                    finish[i] = 1;
                    found = 1;
                }
            }
        }
        if (!found) {
            return 0; // No safe sequence exists
        }
    }
    return 1; // Safe sequence found
}

int main() {
    int n, m; // Number of processes and resources respectively
    printf("Enter number of processes: ");
    scanf("%d", &n);
    printf("Enter number of resources: ");
    scanf("%d", &m);

    // Input maximum resource needs of each process
    int max[MAX_PROCESSES][MAX_RESOURCES];
    printf("Enter maximum resource needs for each process:\n");
    for (int i = 0; i < n; ++i) {
        printf("For Process P%d: ", i);
        for (int j = 0; j < m; ++j)
            scanf("%d", &max[i][j]);
    }

    // Input resources currently allocated to each process
    int allocation[MAX_PROCESSES][MAX_RESOURCES];
    printf("Enter resources currently allocated to each process:\n");
    for (int i = 0; i < n; ++i) {
        printf("For Process P%d: ", i);
        for (int j = 0; j < m; ++j)
            scanf("%d", &allocation[i][j]);
    }

    // Input available resources
    int available[MAX_RESOURCES];
    printf("Enter available resources:\n");
    for (int i = 0; i < m; ++i)
        scanf("%d", &available[i]);

    int safeSeq[MAX_PROCESSES];
```

```
    if (findSafeSequence(n, m, max, allocation, available, safeSeq)) {
        printf("Safe sequence: ");
        for (int i = 0; i < n; ++i)
            printf("P%d ", safeSeq[i]);
        printf("\n");
    } else {
        printf("No safe sequence exists.\n");
    }

    return 0;
}
```