

```

#include <stdio.h>

#define MAX 10
#define RESOURCE_TYPES 3

void calculate_need(int need[MAX][RESOURCE_TYPES], int max[MAX][RESOURCE_TYPES],
                   int allot[MAX][RESOURCE_TYPES], int n) {
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < RESOURCE_TYPES; j++) {
            need[i][j] = max[i][j] - allot[i][j];
        }
    }
}

int is_safe(int processes[], int n, int m, int available[],
            int max[][RESOURCE_TYPES], int allot[][RESOURCE_TYPES]) {
    int need[MAX][RESOURCE_TYPES];
    calculate_need(need, max, allot, n);

    int work[RESOURCE_TYPES];
    int finish[MAX];

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

    for (int i = 0; i < n; i++) {
        finish[i] = 0;
    }

    int safeSeq[MAX];
    int count = 0;

    while (count < n) {
        int found = 0;
        for (int p = 0; p < n; p++) {
            if (!finish[p]) {
                int j;
                for (j = 0; j < m; j++) {
                    if (need[p][j] > work[j]) {
                        break;
                    }
                }
                if (j == m) {
                    for (int k = 0; k < m; k++) {
                        work[k] += allot[p][k];
                    }
                    safeSeq[count++] = p;
                    finish[p] = 1;
                    found = 1;
                }
            }
        }
        if (!found) {
            printf("System is not in a safe state.\n");
            return 0;
        }
    }

    printf("System is in a safe state.\nSafe sequence is: ");
    for (int i = 0; i < n; i++) {
        printf("P%d ", safeSeq[i]);
    }
    printf("\n");
    return 1;
}

int main() {
    int n, m;
    int processes[MAX];
    int available[RESOURCE_TYPES];
    int max[MAX][RESOURCE_TYPES];
    int allot[MAX][RESOURCE_TYPES];

    printf("Enter number of processes: ");
    scanf("%d", &n);

```

```

printf("Enter number of resource types: ");
scanf("%d", &m);

printf("Enter the number of available instances for each resource:\n");
for (int i = 0; i < m; i++) {
    printf("Resource %d: ", i + 1);
    scanf("%d", &available[i]);
}

printf("Enter maximum matrix:\n");
for (int i = 0; i < n; i++) {
    processes[i] = i;
    printf("Process %d:\n", i + 1);
    for (int j = 0; j < m; j++) {
        printf("Maximum resource %d: ", j + 1);
        scanf("%d", &max[i][j]);
    }
}

printf("Enter allocation matrix:\n");
for (int i = 0; i < n; i++) {
    printf("Process %d:\n", i + 1);
    for (int j = 0; j < m; j++) {
        printf("Allocated resource %d: ", j + 1);
        scanf("%d", &allot[i][j]);
    }
}

is_safe(processes, n, m, available, max, allot);

return 0;
}

```