```c
#include <stdio.h>

#include <stdlib.h>

int allocation[20][20], max[20][20], available[20], need[20][20], safe[10], s = 0;

int finish[10], work[10], p, r, i, j, ch, index, req[10];

void check() {

    s = 0;

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

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

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

    printf("\nAllocation Table:\n");

    for (i = 0; i < p; i++) {

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

            printf("%d\t", allocation[i][j]);

        printf("\n");

    }

    printf("\nNeed Table:\n");

    for (i = 0; i < p; i++) {

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

            printf("%d\t", need[i][j]);

        printf("\n");

    }

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

        finish[i] = 0;

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

        work[i] = available[i];

    int executed;

    do {

        executed = 0;

        for (i = 0; i < p; i++) {

            if (finish[i] == 0) {

                int flag = 1;

                for (j = 0; j < r; j++) {

                    if (need[i][j] > work[j]) {

                        flag = 0;

                        break;
                    }
                }
                if (flag) {
                    for (j = 0; j < r; j++)
                        work[j] += allocation[i][j];
                    safe[s++] = i;
                    finish[i] = 1;
                    executed = 1;
                }
            }
        }
    } while (executed);
    for (i = 0; i < p; i++) {
        if (finish[i] == 0) {
            printf("\nSystem is in Deadlock state\n");
            return;
        }
    }
    printf("\nSystem is in Safe state\nSafe Sequence: ");
    for (i = 0; i < p; i++)
        printf("P%d\t", safe[i]);
    printf("\n");
}
int main() {
    printf("\nEnter the number of resources and processes: ");
    scanf("%d%d", &r, &p);
    printf("\nEnter the Allocation Table:\n");
    for (i = 0; i < p; i++)
        for (j = 0; j < r; j++)
            scanf("%d", &allocation[i][j]);
    printf("\nEnter the Max Table:\n");
    for (i = 0; i < p; i++)
        for (j = 0; j < r; j++)
            scanf("%d", &max[i][j]);
    printf("\nEnter the Available vector:\n");
    for (i = 0; i < r; i++)
        scanf("%d", &available[i]);
    check();
    printf("\nDo you want to add a new request? (0/1): ");
    scanf("%d", &ch);
    if (ch == 0)
        exit(0);
    printf("\nEnter the process number: ");
    scanf("%d", &index);
    printf("\nEnter the request: ");
    for (i = 0; i < r; i++)
        scanf("%d", &req[i]);
    for (i = 0; i < r; i++) {
        if (req[i] > need[index][i]) {
            printf("\nRequest cannot be satisfied.\n");
            exit(1);
        }
        if (req[i] > available[i]) {
            printf("\nRequest cannot be satisfied.\n");
            exit(1);
        }
    }
    for (i = 0; i < r; i++) {
        available[i] -= req[i];
        allocation[index][i] += req[i];
        need[index][i] -= req[i];
    }
    printf("\nRequest granted. Rechecking system state...\n");
    check();
    return 0;
}
```