```c
#include <stdio.h>
#include <stdbool.h>

int main() {
    int n, m, i, j, k;

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

    int alloc[n][m], max[n][m], avail[m];

    printf("\nEnter Allocation Matrix:\n");
    for (i = 0; i < n; i++)
        for (j = 0; j < m; j++)
            scanf("%d", &alloc[i][j]);

    printf("\nEnter Maximum Matrix:\n");
    for (i = 0; i < n; i++)
        for (j = 0; j < m; j++)
            scanf("%d", &max[i][j]);

    printf("\nEnter Available Resources:\n");
    for (i = 0; i < m; i++)
        scanf("%d", &avail[i]);
```

```
int f[n], ans[n], ind = 0;

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

    f[k] = 0;


int need[n][m];

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

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

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


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

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

        if (f[i] == 0) {

            bool flag = true;

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

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

                    flag = false;

                    break;

                }

            }


            if (flag) {

                ans[ind++] = i;

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

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

                f[i] = 1;
                }
            }
        }
    }

    bool safe = true;
    for (i = 0; i < n; i++)
        if (f[i] == 0)
            safe = false;

    if (safe) {
        printf("\nSystem is in a Safe State.\nSafe Sequence: ");
        for (i = 0; i < n; i++)
            printf("P%d ", ans[i]);
    } else {
        printf("\nSystem is in an Unsafe State.\n");
    }

    return 0;
}
```