```
#include <stdio.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];
    int need[n][m], finish[n], safeSeq[n];

    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]);

    // Calculate Need Matrix
    for (i = 0; i < n; i++)
        for (j = 0; j < m; j++)
            need[i][j] = max[i][j] - alloc[i][j];
```

```
for (i = 0; i < n; i++)
    finish[i] = 0;

int count = 0;
while (count < n) {
    int found = 0;
    for (i = 0; i < n; i++) {
        if (finish[i] == 0) {
            int flag = 0;
            for (j = 0; j < m; j++) {
                if (need[i][j] > avail[j]) {
                    flag = 1;
                    break;
                }
            }
            if (flag == 0) {
                for (k = 0; k < m; k++)
                    avail[k] += alloc[i][k];
                safeSeq[count++] = i;
                finish[i] = 1;
                found = 1;
            }
        }
    }
    if (found == 0) {
        printf("\nSystem is not in safe state.");
        return 0;
    }
}

printf("\nSystem is in safe state.\nSafe sequence is: ");
```

```
    for (i = 0; i < n; i++)
        printf("P%d ", safeSeq[i]);
    printf("\n");

    return 0;
}
```

bankers.c

```
gcc bankers.c -o bankers
./bankers
```

input

```
Enter number of processes: 5
Enter number of resources: 3

Enter Allocation Matrix:
0 1 0
2 0 0
3 0 2
2 1 1
0 0 2

Enter Maximum Matrix:
7 5 3
3 2 2
9 0 2
4 2 2
5 3 3

Enter Available Resources:
3 3 2
```

output

```
System is in safe state.
Safe sequence is: P1 P3 P4 P0 P2
```