
```

stdio.h>

int main(){
    int n, m, i, j, y, k, ind = 0, flag = 0;

    printf("Enter the Number of Processes: ");
    scanf("%d", &n);
    printf("Enter the Number of Resources: ");
    scanf("%d", &m);

    int max[n][m], allocation[n][m], available[m];

    // Taking input for MAXIMUM Matrix, ALLOCATION Matrix, and Available Vector

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

    printf("Enter the Allocated Matrix of a Process:\n");
    for (i = 0; i < n; i++)
        for (j = 0; j < m; j++)
            scanf("%d", &allocation[i][j]);

    printf("Enter the Available Vector:\n");
    for (i = 0; i < m; i++)
        scanf("%d", &available[i]);

    // Declaring finish array which will represent the TRUE/FALSE value for the Execution
    // of Process
    // and an ans array for storing the SAFE sequence of the processes

    int f[n], ans[n];

    for (i = 0; i < n; i++)
        f[i] = 0; // Initialize f[] to 0

    // Calculating the NEED Matrix from the MAXIMUM and ALLOCATION Matrix which is NEED =
    // MAX - ALLOCATION

    int need[n][m];

    for (i = 0; i < n; i++)
        for (j = 0; j < m; j++)
            need[i][j] = max[i][j] - allocation[i][j];

    // Printing ALLOCATION Matrix, NEED Matrix, and Available vector

    printf("\nPrinting ALLOCATION Matrix:\n");
    for (i = 0; i < n; i++) {
        for (j = 0; j < m; j++) {
            printf("%d\t", allocation[i][j]);
        }
        printf("\n");
    }

    printf("\nPrinting NEED Matrix:\n");
    for (i = 0; i < n; i++) {
        for (j = 0; j < m; j++) {
            printf("%d\t", need[i][j]);
        }
        printf("\n");
    }

    printf("\nPrinting AVAILABLE VECTOR:\n");

```

```

    for (i = 0; i < m; i++)
        printf("%d\t", available[i]);

    printf("\n\n");

    // Implementing the Banker's Algorithm

    for (k = 0; k < n; k++) {
        for (i = 0; i < n; i++) {
            if (f[i] == 0) {
                flag = 0;
                for (j = 0; j < m; j++) {
                    if (need[i][j] > available[j]) {
                        flag = 1;
                        break;
                    }
                }
                if (flag == 0) {
                    ans[ind++] = i;
                    for (y = 0; y < m; y++)
                        available[y] += allocation[i][y];
                    f[i] = 1;
                }
            }
        }
    }

    flag = 1;
    for (i = 0; i < n; i++) {
        if (f[i] == 0) {
            flag = 0;
            printf("\nFollowing System is NOT in Safe Sequence\n");
            break;
        }
    }

    if (flag == 1) {
        printf("\nFollowing System is in SAFE STATE\n\n");
        printf("The Following is the SAFE SEQUENCE that will be executed according to the\n\n");
        printf("Banker's Algorithm:\n\n");
        for (i = 0; i < n - 1; i++) {
            printf("P%d -> ", ans[i]);
        }
        printf("P%d\n", ans[n - 1]);
    }

    return 0;
}

```