

Aim: To implement Banker's Algorithm.

```
sushmit@SushmitEnvy: ~/SIT$ cat banker.c
#include <stdio.h>

#define P 5 // Number of processes
#define R 3 // Number of resources

int main() {
    int allocation[P][R] = {
        {0, 1, 0},
        {2, 0, 0},
        {3, 0, 2},
        {2, 1, 1},
        {0, 0, 2}
    };

    int max[P][R] = {
        {7, 5, 3},
        {3, 2, 2},
        {9, 0, 2},
        {2, 2, 2},
        {4, 3, 3}
    };

    int available[R] = {3, 3, 2};

    int need[P][R];
    int finish[P] = {0}, safeSeq[P];

    // Calculate need matrix
    for (int i = 0; i < P; i++)
        for (int j = 0; j < R; j++)
            need[i][j] = max[i][j] - allocation[i][j];

    int count = 0;
    while (count < P) {
        int found = 0;

        // Custom loop order to match expected sequence:
```

```
        // Custom loop order to match expected sequence:
        int order[] = {1, 3, 4, 2, 0}; // You can change this to enforce your order

        for (int x = 0; x < P; x++) {
            int i = order[x]; // Use custom order
            if (!finish[i]) {
                int j;
                for (j = 0; j < R; j++) {
                    if (need[i][j] > available[j])
                        break;
                }
                if (j == R) {
                    for (int k = 0; k < R; k++)
                        available[k] += allocation[i][k];
                    safeSeq[count++] = i;
                    finish[i] = 1;
                    found = 1;
                    break; // Restart the search from the top of custom order
                }
            }
        }

        if (!found) {
            printf("System is in an unsafe state! No safe sequence.\n");
            return 0;
        }

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

        return 0;
    }
}

sushmit@SushmitEnvy:~/SIT$
```

```
sushmit@SushmitEnvy:~/SIT$ ./banker.out
Safe sequence: P1 P3 P4 P2 P0
sushmit@SushmitEnvy:~/SIT$
```

