Assignment 5

```
#include <stdio.h>

int main() {
   int n, m; // n = number of processes, m = number of resources
   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];
   int safeSeq[n], index = 0;

   // Input Allocation Matrix
   printf("Enter allocation matrix:\n");
   for (int i = 0; i < n; i++)
      for (int j = 0; j < m; j++)
         scanf("%d", &alloc[i][j]);

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

   // Input Available Resources
   printf("Enter available resources:\n");
   for (int i = 0; i < m; i++)
      scanf("%d", &avail[i]);

   // Calculate Need Matrix = Max - Allocation
   printf("\nNeed matrix:\n");
   for (int i = 0; i < n; i++) {
      for (int j = 0; j < m; j++) {
         need[i][j] = max[i][j] - alloc[i][j];
         printf("%d ", need[i][j]);
      }
      printf("\n");
   }

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

   // Find Safe Sequence
   for (int k = 0; k < n; k++) { // repeat until all processes are checked
      for (int i = 0; i < n; i++) {
         if (finish[i] == 0) {
            int canExecute = 1;
```

```
            for (int j = 0; j < m; j++) {
                if (need[i][j] > avail[j]) {
                    canExecute = 0;
                    break;
                }
            }

            if (canExecute) {
                // Add to safe sequence
                for (int j = 0; j < m; j++)
                    avail[j] += alloc[i][j];

                safeSeq[index++] = i;
                finish[i] = 1;
            }
        }
    }
  }

  // Check if all processes could finish
  int safe = 1;
  for (int i = 0; i < n; i++)
    if (finish[i] == 0)
      safe = 0;

  if (safe) {
    printf("\nSystem is in a SAFE state.\nSafe sequence: ");
    for (int i = 0; i < n; i++)
      printf("P%d ", safeSeq[i]);
  } else {
    printf("\nSystem is in an UNSAFE state (deadlock possible).\n");
  }

  return 0;
}
```

Code:

```
manasvi@manasvi:/mnt/c/Users/bhute/Desktop/oslab/Ass5$ gcc ass5.c -o banker
manasvi@manasvi:/mnt/c/Users/bhute/Desktop/oslab/Ass5$ ./banker
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 max matrix:
7 5 3
3 2 2
9 0 2
2 2 2
4 3 3
Enter available resources:
3 3 2

Need matrix:
7 4 3
1 2 2
6 0 0
0 1 1
4 3 1

System is in a SAFE state.
Safe sequence: P1 P3 P4 P0 P2 manasvi@manasvi:/mnt/c/Users/bhute/Desktop/os
```