**Ex. No.: 9**
**Date:**

## DEADLOCK AVOIDANCE

**Aim:**

To find out a safe sequence using Banker's algorithm for deadlock avoidance.

**Algorithm:**

1. Initialize work=available and finish[i]=false for all values of i
2. Find an i such that both:
 finish[i]=false and $Need_i$<= work
3. If no such i exists go to step 6
4. Compute work=work+allocationi
5. Assign finish[i] to true and go to step 2
6. If finish[i]==true for all i, then print safe sequence
7. Else print there is no safe sequence

**Program Code:**

```
#include <stdio.h>

int main() {
   int n = 5, m = 3;
   int alloc[5][3] = {
      { 0, 1, 0 },
      { 2, 0, 0 },
      { 3, 0, 2 },
      { 2, 1, 1 },
      { 0, 0, 2 }
   };

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

   int avail[3] = { 3, 3, 2 };

   int f[n], ans[n], ind = 0;
   for (int k = 0; k < n; k++) f[k] = 0;

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

   int y = 0;
   for (int k = 0; k < 5; k++) {
      for (int i = 0; i < n; i++) {
         if (f[i] == 0) {
            int flag = 0;
```

```
            for (int j = 0; j < m; j++) {
                if (need[i][j] > avail[j]) {
                    flag = 1;
                    break;
                }
            }

            if (flag == 0) {
                ans[ind++] = i;
                for (int y = 0; y < m; y++)
                    avail[y] += alloc[i][y];
                f[i] = 1;
            }
        }
    }
}

printf("The SAFE Sequence is \n");
for (int i = 0; i < n - 1; i++)
    printf("P%d -> ", ans[i]);
printf("P%d", ans[n - 1]);

return 0;
}
```

**Sample Output:**

The SAFE Sequence is
P1 -> P3 -> P4 -> P0 -> P2

**Output:**

```c
#include <stdio.h>
int main() {

    int n = 5, m = 3;

    int alloc[5][3] = {

        { 0, 1, 0 },

        { 2, 0, 0 },

        { 3, 0, 2 },

        { 2, 1, 1 },

        { 0, 0, 2 }

    };


    int max[5][3] = {
```

Input

```
The SAFE Sequence is
P1 -> P3 -> P4 -> P0 -> P2

...Program finished with exit code 0
Press ENTER to exit console.
```

**Result:**
Program is executed successfully and output is verified.