OS Lab Assignment

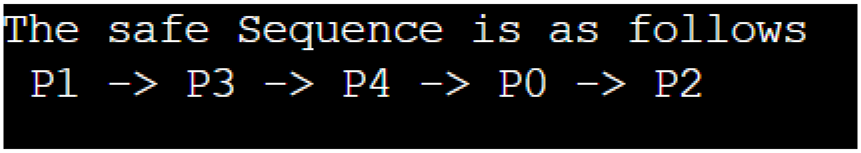
1.) To write a C/C++ program to implement Banker's algorithm for deadlock avoidance.

A.) CODE:-

```
#include <stdio.h>
int main()
{
        int n, r, i, j, k;
        n = 5;
        r = 3;
        int alloc[5][3] = { { 0, 0, 1 },
                                        { 3, 0, 0 },
                                        { 1, 0, 1 },
                                        { 2, 3, 2 },
                                        { 0, 0, 3 } };
        int max[5][3] = { { 7, 6, 3 },
                                { 3, 2, 2 },
                                { 8, 0, 2 },
                                { 2, 1, 2 },
                                { 5, 2, 3 } };
        int avail[3] = { 2, 3, 2 };
        int f[n], ans[n], ind = 0;
        for (k = 0; k < n; k++) {
                f[k] = 0;
        }
        int need[n][r];
        for (i = 0; i < n; i++) {
                for (j = 0; j < r; j++)
                        need[i][j] = max[i][j] - alloc[i][j];
        }
```

```
    int y = 0;
    for (k = 0; k < 5; k++) {
        for (i = 0; i < n; i++) {
            if (f[i] == 0) {
                int flag = 0;
                for (j = 0; j < r; j++) {
                    if (need[i][j] > avail[j]){
                        flag = 1;
                        break;
                    }
                }
                if (flag == 0) {
                    ans[ind++] = i;
                    for (y = 0; y < r; y++)
                        avail[y] += alloc[i][y];
                    f[i] = 1;
                }
            }
        }
    }
    printf("The safe Sequence is as follows\n");
    for (i = 0; i < n - 1; i++)
        printf(" P%d ->", ans[i]);
    printf(" P%d", ans[n - 1]);
    return (0);
}
```

SAMPLE INPUT AND SAMPLE OUTPUT:



```
The safe Sequence is as follows
 P1 -> P3 -> P4 -> P0 -> P2
```