//Implement the C program for Deadlock Avoidance Algorithm: Bankers Algorithm.

// Banker's Algorithm

#include <stdio.h>

int main()

{

    // P0, P1, P2, P3, P4 are the Process names here

    int n, m, i, j, k;

    n = 5; // Number of processes

    m = 3; // Number of resources

    int alloc[5][3] = { { 0, 0, 1 }, // P0    // Allocation Matrix

                        { 2, 0, 0 }, // P1

                        { 1, 3, 2 }, // P2

                        { 1,0, 1 }, // P3

                        { 0, 0, 1 } }; // P4

    int max[5][3] = { { 1, 1, 1 }, // P0    // MAX Matrix

                      { 3, 2, 3 }, // P1

                      { 4, 3, 1 }, // P2

                      { 0, 0, 1 }, // P3

                      { 3, 2,1 } }; // P4

    int avail[3] = { 3, 0, 1 }; // Available Resources

    int f[n], ans[n], ind = 0;

    for (k = 0; k < n; k++) {

        f[k] = 0;

    }

    int need[n][m];

    for (i = 0; i < n; i++) {

        for (j = 0; j < m; 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 < m; j++) {

                    if (need[i][j] > avail[j]){

                        flag = 1;

                         break;

                    }

                }

                if (flag == 0) {

                    ans[ind++] = i;

                    for (y = 0; y < m; y++)

                        avail[y] += alloc[i][y];

                    f[i] = 1;

                }

            }

        }

    }

      int flag = 1;

      for(int i=0;i<n;i++)

    {

      if(f[i]==0)

      {

        flag=0;

         printf("The following system is not safe");

        break;

      }

    }

      if(flag==1)

    {

      printf("Following is the SAFE Sequence\n");

      for (i = 0; i < n - 1; i++)

        printf(" P%d ->", ans[i]);

      printf(" P%d", ans[n - 1]);

    }

    return 0;

}