BANKER'S ALGORITHM

```c
// Banker's Algorithm
#include <stdio.h>
int alloc[10][10],avail[3],need[10][10],max[10][10];
int n=5,m=3;
void banker(){
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 (y = 0; y < m; y++)
```

```c
                                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 (int i = 0; i < n - 1; i++)
                printf(" P%d ->", ans[i]);
        printf(" P%d", ans[n - 1]);
        }
}
int main()
{
        int c,pro,res[3];
    printf("Enter the allocations\n");
        int  i, j, k;
        for(i=0;i<5;i++){
        for(j=0;j<3;j++){
           scanf("%d",&alloc[i][j]);
```

```c
        }
       }
      printf("Enter the max\n");
      for(i=0;i<5;i++){
     for(j=0;j<3;j++){
           scanf("%d",&max[i][j]);
      }
       }

      printf("Enter the available resources\n");
      for(int i=0;i<3;i++)
     scanf("%d",&avail[i]);
      banker();
      return (0);

}
```

OUTPUT:

```
Enter the allocations
0 1 0
2 0 0
3 0 2
2 1 1
0 0 2
Enter the max
7 5 3
3 2 2
9 0 2
2 2 2
4 3 3
Enter the available resources
3 3 2
Following is the SAFE Sequence
 P1 -> P3 -> P4 -> P0 -> P2
Process returned 0 (0x0)   execution time : 50.368 s
Press any key to continue.
```