CODE:

```
#include <stdio.h>
#include <stdbool.h>
#define Pro 5
#define Res 3
 bool isafestate(int processes[],int avail[],int max[][Res],int allot[][Res],int n,int m){
        int need[n][m];
        bool finish[n];
        int safeSeq[n];
        int work[m];
        int i,j;
for (i = 0; i < n; i++){
        for (j = 0; j < m; j++){
                need[i][j] = max[i][j] - allot[i][j];
            }
        }
for (i = 0; i < m; i++){
            work[i] = avail[i];
        }
for (i = 0; i < n; i++){
            finish[i] = false;
        }
        int count = 0;
while ( count < n){
            bool found = false;
```

```
int p,k;


for ( p = 0;p < n; p++){
        if(!finish[p]) {
                int j;
                for (j = 0;j < m;j++){
                        if (need[p][j] > work[j]){
                                break;
                        }
                }
                if(j == m){
                        for(k =0;k < m; k++){
                                work[k] +=allot[p][k];
                        }
                        safeSeq[count++] = p;
                        finish[p] = true;
                        found = true;
                        break;
                }
        }
}
if (!found){
        printf("System is in an unsafe state.\n");
        return false;
}
```

```
        }
        printf("System is in a safe state.\n Safe sequence is: ");
        for(i = 0; i < n; i++){
                printf("%d\t",safeSeq[i]);
        }
        printf("\n");
        return true;
}
int main(){
        int processes[] = {0 , 1, 2, 3, 4};
        int avail[] = {5, 3, 2};

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

        int allot[][Res] = {            {0, 1, 0},
                                        {2, 0, 0},
                                        {3, 0, 2},
                                        {2, 1, 1},
                                        {0, 0, 2},
                         };
```

```
    isafestate(processes, avail, max, allot, Pro, Res);
    return 0;
}
```

OUTPUT: