**BANKER’S ALGORITHM**

**PROGRAM**

#include <stdio.h>

int main()

{ int numResources, numProcesses ,sum , index=0;

printf("\nEnter the number of processes: ");

scanf("%d", &numProcesses);

printf("\nEnter the number of resources: ");

scanf("%d",&numResources);

int available[numResources], allocation[numProcesses][numResources], max[numProcesses][numResources],

need[numProcesses][numResources],processCompleted[numProcesses],safeSequence[numProcesses];

printf("\nEnter the Available Resources\n");

for (int j = 0; j < numResources; j++)

scanf("%d", &available[j]);

printf("\nEnter the Allocated Matrix:\n");

for (int i = 0; i < numProcesses; i++)

for (int j = 0; j < numResources; j++)

scanf("%d", &allocation[i][j]);

printf("\nEnter the Max Matrix:\n");

for (int i = 0; i < numProcesses; i++)

for (int j = 0; j < numResources; j++)

{

scanf("%d", &max[i][j]);

need[i][j] = max[i][j] - allocation[i][j];

}

printf("\nThe Need Matrix is:\n");

for (int i = 0; i < numProcesses; i++)

{for (int j = 0; j < numResources; j++)

printf("%d ", need[i][j]);

printf("\n");

processCompleted[i] = 0;

}

for (int k = 0; k < numProcesses; k++)

for (int i = 0; i < numProcesses; i++)

if (processCompleted[i] == 0)

{

int flag = 0;

for (int j = 0; j < numResources; j++)

if (need[i][j] > available[j])

{

flag = 1;

break;

}

if (flag == 0)

{

safeSequence[index++] = i;

for (int l = 0; l < numResources; l++)

available[l]+= allocation[i][l];

processCompleted[i] = 1;

}

}

for (int i = 0; i < numProcesses; i++)

if (processCompleted[i] == 0)

{

printf("The system is in unsafe state.");

return 0;

}

printf("\nThe SAFE Sequence is\n");

for (int i = 0; i < numProcesses - 1; i++)

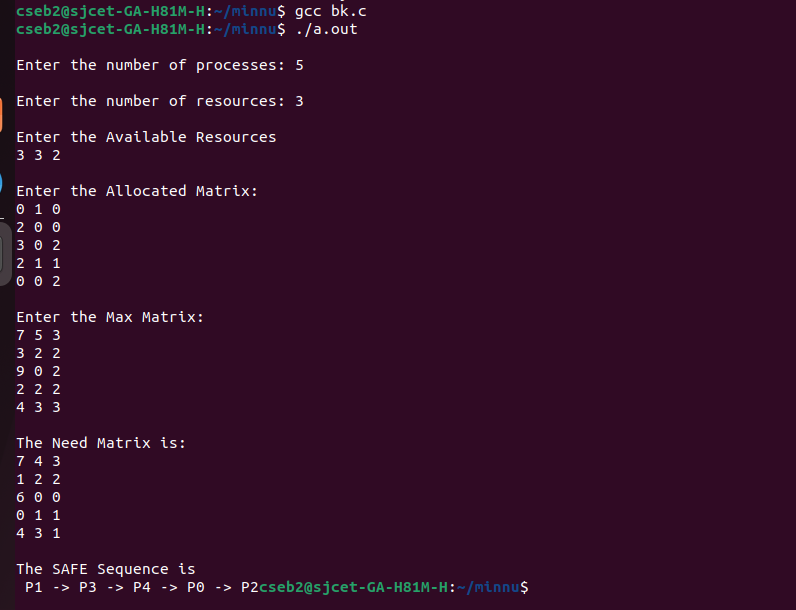
printf(" P%d ->", safeSequence[i]);

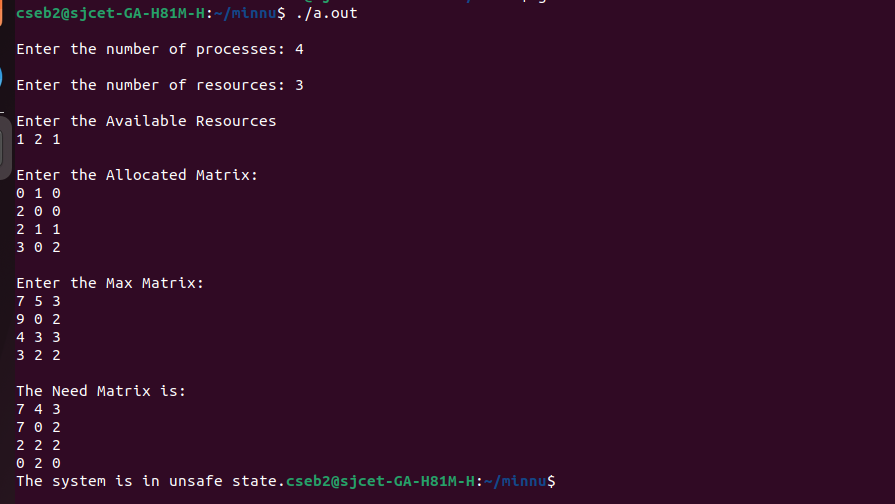
printf(" P%d", safeSequence[numProcesses - 1]);

return 0;

}

**OUTPUT**

****

****