**LAB 10 ASSIGNMENT**

**Name:** Birva Babaria

**Roll no.:** CE010

**ID:** 19CEUON064

**Aim:** Include the code and proper output screenshots of the implementation of Safety Algorithm and Resource Request Algorithm.

**1). Write a program for ‘Safety Algorithm’.**

**CODE:**

//Safety algorithm

#include <stdio.h>

int check(int i, int resource, int need[][resource], int work[])

{

int k = 0;

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

{

if (need[i][j] <= work[j])

{

++k;

}

}

return k;

}

void SafetyAlgo(int process, int resource, int allocation[][resource], int max[][resource], int available[])

{

int finish[process], work[resource], order[process], need[process][resource];

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

{

finish[i] = 0;

}

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

{

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

{

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

}

}

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

{

work[i] = available[i];

}

int k = 0;

for (int p = 0; p < 5; p++)

{

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

{

if (finish[i] == 0 && check(i, resource, need, work) == resource)

{

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

{

work[j] = work[j] + allocation[i][j];

}

finish[i] = 1;

order[k++] = i;

}

}

}

int p = 0;

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

{

if (finish[i] == 1)

{

++p;

}

}

printf("Allocation:\n");

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

{

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

{

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

}

printf("\n");

}

printf("Need:\n");

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

{

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

{

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

}

printf("\n");

}

printf("Available:\n");

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

{

printf("%d ", available[j]);

}

printf("\n");

if (k == process)

{

printf("Safe Sequence:\n");

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

{

printf("P%d ", order[i]);

}

printf("\n");

}

if (p == process)

{

printf("System in safe state\n");

}

else

{

printf("System not in safe state\n");

}

}

int main()

{

int process, resource;

printf("Enter Number of Process:\n");

scanf("%d", &process);

printf("Enter Number of Resource:\n");

scanf("%d", &resource);

int allocation[process][resource],

max[process][resource], available[resource];

printf("Enter Allocation :\n");

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

{

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

{

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

}

}

printf("Enter Max :\n");

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

{

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

{

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

}

}

printf("Enter Available:\n");

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

{

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

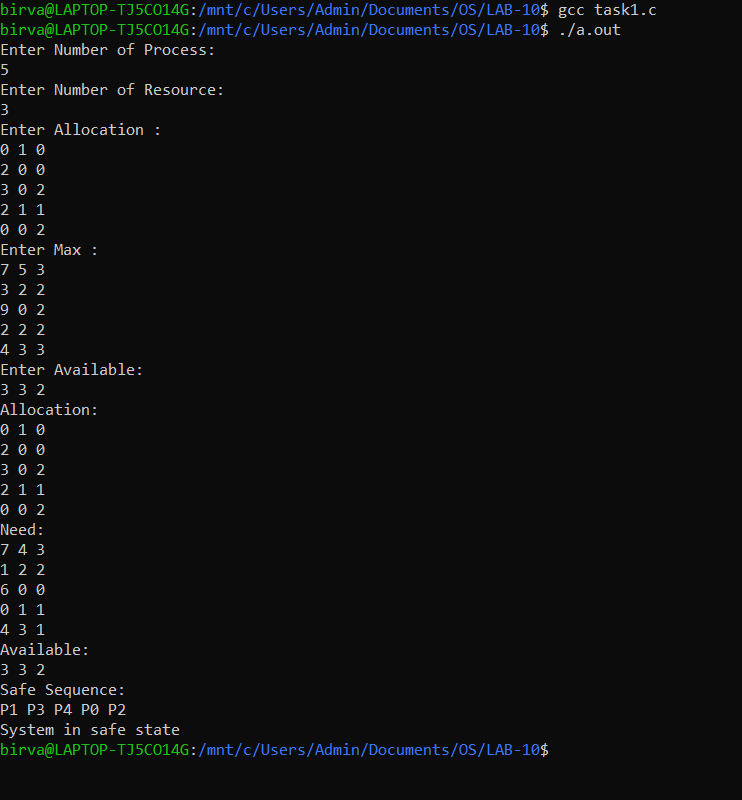
}

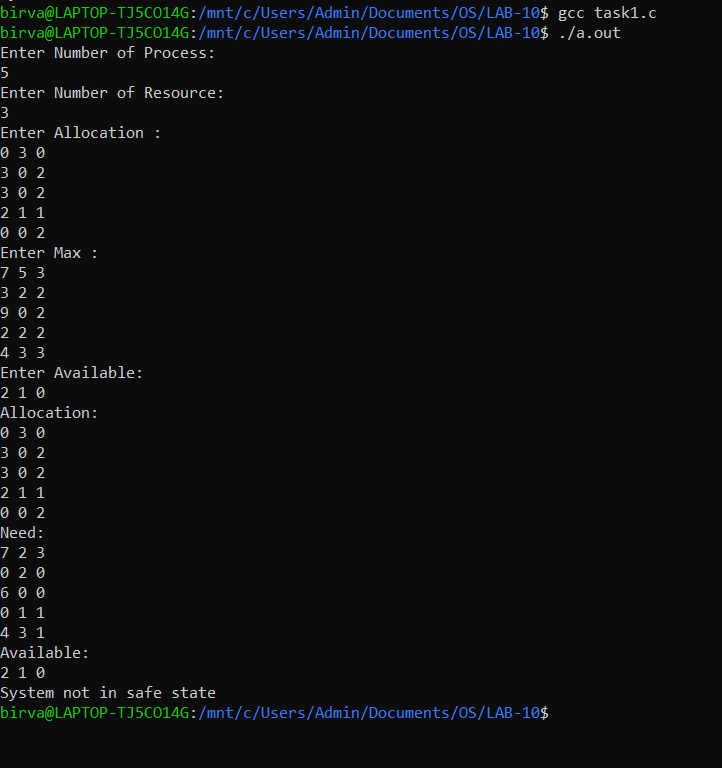
SafetyAlgo(process, resource, allocation, max, available);

return 0;

}

**OUTPUT:**

****



**2). Write a program for ‘Resource request Algorithm’.**

**CODE:**

//Resource request algorithm

#include <stdio.h>

#include<stdlib.h>

int check(int i,int resource,int need[][resource],int work[])

{

int k = 0;

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

{

if (need[i][j] <= work[j])

{

++k;

}

}

return k;

}

void SafetyAlgo(int process,int resource,int allocation[][resource],int max[][resource],int need[][resource],int available[])

{

int finish[process] , work[resource], order[process];

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

{

finish[i]=0;

}

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

{

work[i] = available[i];

}

int k = 0;

for (int p = 0; p < 10; p++)

{

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

{

if (finish[i] == 0 && check(i,resource,need,work) == resource)

{

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

{

work[j] = work[j] + allocation[i][j];

}

finish[i] = 1;

order[k] = i;

k++;

}

}

}

if (k == process)

{

printf("Safe Sequence:\n");

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

{

printf("P%d ", order[i]);

}

printf("\n");

}

int p = 0;

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

{

if (finish[i] == 1)

{

++p;

}

}

if (p == process)

{

printf("System in safe state\n");

}

else

{

printf("System not in safe state\n");

}

}

void RequestResource(int process,int resource,int allocation[][resource],int max[][resource],int available[],int Request[],int processno)

{

int need[process][resource];

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

{

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

{

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

}

}

int k = 0;

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

{

if (Request[j] <= need[processno][j])

{

++k;

}

}

if (k == resource)

{

int p = 0;

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

{

if (Request[j] <= available[j])

{

++p;

}

}

if (p == resource)

{

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

{

available[j] = available[j] - Request[j];

allocation[processno][j] = allocation[processno][j] + Request[j];

need[processno][j] = need[processno][j] - Request[j];

}

}

else

{

printf("Process P%d must Wait for resource because currently resource is not available\n",processno);

exit(0);

}

}

else

{

printf("Error: Request is greater than need\n");

exit(0);

}

printf("Allocation:\n");

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

{

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

{

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

}

printf("\n");

}

printf("Need:\n");

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

{

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

{

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

}

printf("\n");

}

printf("Available:\n");

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

{

printf("%d ", available[j]);

}

printf("\n");

printf("Process no is P%d\n", processno);

printf("Request:\n");

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

{

printf("%d ", Request[j]);

}

printf("\n");

SafetyAlgo(process,resource,allocation,max,need,available);

}

int main()

{

int process,resource;

printf("Enter Number of Process:\n");

scanf("%d",&process);

printf("Enter Number of Resource:\n");

scanf("%d",&resource);

int allocation[process][resource],

max[process][resource],available[resource],Request[resource], processno;

printf("Enter Allocation :\n");

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

{

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

{

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

}

}

printf("Enter Max :\n");

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

{

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

{

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

}

}

printf("Enter Available:\n");

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

{

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

}

printf("Enter Request:\n");

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

{

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

}

printf("Enter process number request for resource:\n");

scanf("%d", &processno);

RequestResource(process,resource,allocation,max,available,Request,processno);

return 0;

}

**OUTPUT:**

