Name : Manasi Rathod

Roll No : 23552

* **CODE**

#include <stdio.h>

#include <stdbool.h>

#define MAX\_PROCESSES 100

#define MAX\_RESOURCES 100

int available[MAX\_RESOURCES];

int maximum[MAX\_PROCESSES][MAX\_RESOURCES];

int allocation[MAX\_PROCESSES][MAX\_RESOURCES];

int need[MAX\_PROCESSES][MAX\_RESOURCES];

bool finish[MAX\_PROCESSES];

int num\_processes, num\_resources;

bool is\_safe\_state();

int main()

{

printf("Enter the number of processes: ");

scanf("%d", &num\_processes);

printf("Enter the number of resources: ");

scanf("%d", &num\_resources);

printf("Enter the available resources:\n");

for (int i = 0; i < num\_resources; i++) {

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

}

printf("Enter the maximum resource allocation for each process:\n");

for (int i = 0; i < num\_processes; i++) {

printf("Process %d: ", i);

for (int j = 0; j < num\_resources; j++) {

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

need[i][j] = maximum[i][j];

}

}

printf("Enter the current resource allocation for each process:\n");

for (int i = 0; i < num\_processes; i++) {

printf("Process %d: ", i);

for (int j = 0; j < num\_resources; j++) {

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

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

available[j] -= allocation[i][j];

}

}

if (is\_safe\_state()) {

printf("The system is in a safe state.\n");

} else {

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

}

return 0;

}

bool is\_safe\_state()

{

int work[MAX\_RESOURCES];

bool found;

for (int i = 0; i < num\_resources; i++) {

work[i] = available[i];

}

for (int i = 0; i < num\_processes; i++) {

finish[i] = false;

}

int count = 0;

while (count < num\_processes) {

found = false;

for (int i = 0; i < num\_processes; i++) {

if (!finish[i]) {

int j;

for (j = 0; j < num\_resources; j++) {

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

break;

}

}

if (j == num\_resources) {

for (int k = 0; k < num\_resources; k++) {

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

}

finish[i] = true;

found = true;

count++;

}

}

}

if (!found) {

break;

}

}

if (count == num\_processes) {

return true;

} else {

return false;

}

}

* **OUTPUT**

