1.Consider a system with 4 processes and 3 resources with the given resource matrices. Claim matrix Allocation matrix 3 2 2 1 0 0 6 1 3 6 1 2 3 1 4 2 1 1 4 2 2 0 0 2 The resource vector is [9,3,6]. Write a C program to determine if the system is in safe or unsafe state.

#include <stdio.h>

#include <stdbool.h>

#define NUM\_PROCESSES 4

#define NUM\_RESOURCES 3

int main() {

int claim[NUM\_PROCESSES][NUM\_RESOURCES] = {{3, 2, 2}, {6, 1, 3}, {3, 1, 4}, {4, 2, 2}};

int allocation[NUM\_PROCESSES][NUM\_RESOURCES] = {{1, 0, 0}, {6, 1, 2}, {2, 1, 1}, {0, 0, 2}};

int available[NUM\_RESOURCES] = {9, 3, 6};

bool finished[NUM\_PROCESSES] = {false};

int work[NUM\_RESOURCES];

for (int i = 0; i < NUM\_RESOURCES; i++) {

work[i] = available[i];

}

int safe\_sequence[NUM\_PROCESSES];

int num\_safe = 0;

while (num\_safe < NUM\_PROCESSES) {

bool found = false;

for (int i = 0; i < NUM\_PROCESSES; i++) {

if (!finished[i]) {

bool can\_finish = true;

for (int j = 0; j < NUM\_RESOURCES; j++) {

if (claim[i][j] - allocation[i][j] > work[j]) {

can\_finish = false;

break;

}

}

if (can\_finish) {

for (int j = 0; j < NUM\_RESOURCES; j++) {

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

}

finished[i] = true;

safe\_sequence[num\_safe++] = i;

found = true;

}

}

}

if (!found) {

printf("Unsafe state!\n");

return 1;

}

}

printf("Safe sequence: ");

for (int i = 0; i < NUM\_PROCESSES; i++) {

printf("%d ", safe\_sequence[i]);

}

printf("\n");

return 0;

}

