#include <stdio.h>

int main() {

int processes = 4;

int resources = 3;

int available[resources] = {9, 3, 6};

int claim[processes][resources] = {

{3, 2, 2},

{6, 1, 3},

{3, 1, 4},

{4, 2, 2}

};

int allocation[processes][resources] = {

{1, 0, 0},

{6, 1, 2},

{2, 1, 1},

{0, 0, 2}

};

int work[resources];

int finish[processes];

int i, j, k, safe, count;

// Initialize work to available

for (i = 0; i < resources; i++) {

work[i] = available[i];

}

// Initialize finish to false for all processes

for (i = 0; i < processes; i++) {

finish[i] = 0;

}

// Loop until all processes are finished or a safe sequence is found

safe = 0;

count = 0;

while (!safe && count < processes) {

safe = 1;

for (i = 0; i < processes; i++) {

if (!finish[i]) {

// Check if process i can be completed with current resources

for (j = 0; j < resources; j++) {

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

safe = 0;

break;

}

}

if (safe) {

// Mark process i as finished and release its resources

finish[i] = 1;

for (k = 0; k < resources; k++) {

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

}

count++;

}

}

}

}

Output:

System is in a safe state

--------------------------------

Process exited after 0.04931 seconds with return value 0

Press any key to continue . . .