**Assignment Number 05**

**Name:** Mihir Unmesh Patil

**Roll NO:** TYCOC213

**Batch**: C/ C-3

**CODE:**

#include <stdio.h>

#include <stdbool.h>

int main() {

int n, m;

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

scanf("%d", &n);

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

scanf("%d", &m);

int allocation[n][m], max[n][m], need[n][m], available[m];

bool finish[n];

printf("Enter the allocation matrix:\n");

for (int i = 0; i < n; i++) {

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

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

}

}

printf("Enter the max matrix:\n");

for (int i = 0; i < n; i++) {

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

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

}

}

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

for (int i = 0; i < m; i++) {

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

}

for (int i = 0; i < n; i++) {

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

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

finish[i] = false;

}

}

int safeSequence[n];

int count = 0;

int work[m];

for (int i = 0; i < m; i++) {

work[i] = available[i];

}

while (count < n) {

bool found = false;

for (int i = 0; i < n; i++) {

if (finish[i] == false) {

int j;

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

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

break;

}

}

if (j == m) {

for (int k = 0; k < m; k++) {

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

}

safeSequence[count++] = i;

finish[i] = true;

found = true;}}}

if (found == false) {

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

return 0;

}

}

printf("System is in a safe state. Safe sequence is: ");

for (int i = 0; i < n; i++) {

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

}

printf("\n");

return 0;

}

**Output:**

