**Name: Atharva Salitri Division: CSAI B**

**Roll No.: 37 PRN: 12310120**

**Subject: OS Lab Assignment 6**

**Title: Implementing Deadlock Detection**

**1)Deadlock Detection**

**Code:**

#include <stdio.h>

int main() {

int process, resource, i, j, c = 0, count1 = 0, count2 = 0;

printf("Enter number of processes: ");

scanf("%d", &process);

printf("Enter number of Resources: ");

scanf("%d", &resource);

int avail[resource], max[process][resource], allot[process][resource], need[process][resource], completed[process];

int safeSequence[process]; // To store the safe sequence

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

completed[i] = 0;

printf("Enter available resource instances: ");

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

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

}

printf("\nEnter max needs matrix: ");

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

printf("\nFor P[%d]: ", i);

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

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

}

}

printf("\nEnter allocated matrix: ");

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

printf("\nFor P[%d]: ", i);

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

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

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

}

}

while (count1 != process) {

count2 = count1;

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

if (completed[i] == 0) { // Process not completed

c = 0;

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

if (need[i][j] <= avail[j]) {

c++;

}

}

if (c == resource) { // Process can proceed

completed[i] = 1; // Mark process as completed

safeSequence[count1++] = i; // Store the safe sequence

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

avail[j] += allot[i][j]; // Release allocated resources

}

}

}

}

if (count1 == count2) { // No progress made, indicating a deadlock

printf("\nDeadlock Detected\n");

return 0;

}

}

printf("\nNo Deadlock Detected\n");

printf("Safe Sequence: ");

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

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

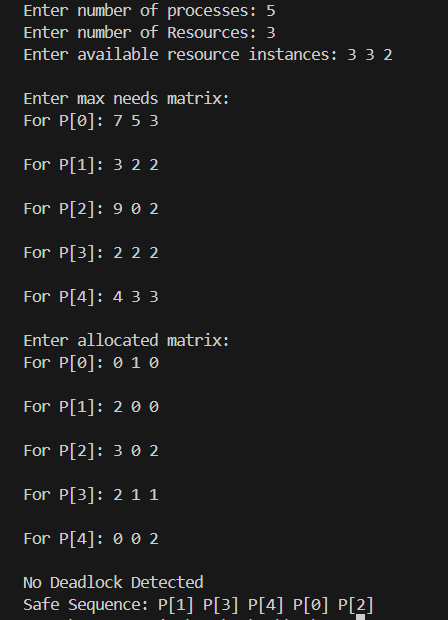
}

printf("\n");

return 0;

}

**Output:**

****