Bansilal Ramnath Agarwal Charitable Trust's

Vishwakarma Institute of Technology, Pune-37

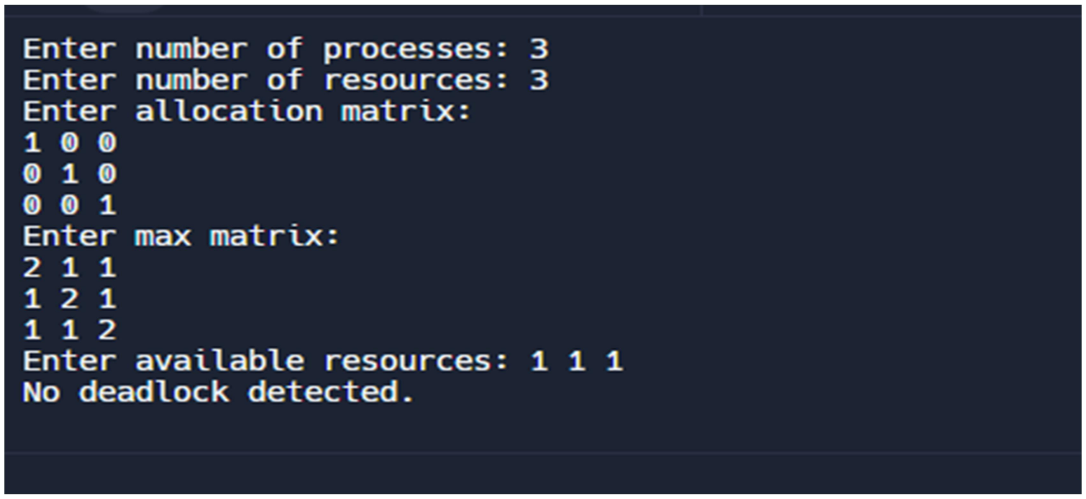
(Anautonomous Institute of Savitribai Phule Pune University)

Department of Computer Engineering

Division	**CS**
Batch	**B1**
Roll no.	**90**
Name	**Aditya Shrinivas Kurapati**
PRN No	**12320184**

- **Deadlock Detection, Deadlock Recovery**

```
#include <stdbool.h>
#include <stdio.h>

#define MAX_PROCESSES 10
#define MAX_RESOURCES 10

int allocation[MAX_PROCESSES][MAX_RESOURCES];
int max[MAX_PROCESSES][MAX_RESOURCES];
int available[MAX_RESOURCES];
bool marked[MAX_PROCESSES];
int num_processes, num_resources;

bool isDeadlock() {
  bool visited[MAX_PROCESSES] = {false};

  for (int i = 0; i < num_processes; ++i) {
    if (!marked[i] && !visited[i]) {
      int work[MAX_RESOURCES];
      for (int j = 0; j < num_resources; ++j) {
        work[j] = available[j];
      }

      bool finish = false;
      while (!finish) {
        finish = true;
        for (int j = 0; j < num_processes; ++j) {
          if (!visited[j] && !marked[j]) {
            bool can_allocate = true;
            for (int k = 0; k < num_resources; ++k) {
              if (max[j][k] - allocation[j][k] > work[k]) {
                can_allocate = false;
                break;
              }
            }
            if (can_allocate) {
              for (int k = 0; k < num_resources; ++k) {
                work[k] += allocation[j][k];
              }
              visited[j] = true;
              finish = false;
            }
          }
        }
      }

      for (int j = 0; j < num_processes; ++j) {
        if (!visited[j] && !marked[j]) {
          return true; // Deadlock detected
        }
      }
    }
  }
  return false; // No deadlock detected
}

int main() {
  printf("Enter number of processes: ");
  scanf("%d", &num_processes);
  printf("Enter number of resources: ");
  scanf("%d", &num_resources);

  printf("Enter allocation matrix:\n");
  for (int i = 0; i < num_processes; ++i) {
    for (int j = 0; j < num_resources; ++j) {
      scanf("%d", &allocation[i][j]);
    }
  }
  printf("Enter max matrix:\n");
  for (int i = 0; i < num_processes; ++i) {
    for (int j = 0; j < num_resources; ++j) {
      scanf("%d", &max[i][j]);
    }
  }

  printf("Enter available resources: ");
  for (int i = 0; i < num_resources; ++i) {
    scanf("%d", &available[i]);
  }

  if (isDeadlock()) {
    printf("Deadlock detected!\n");
  } else {
    printf("No deadlock detected.\n");
  }

  return 0;
}
```



```
Enter number of processes: 3
Enter number of resources: 3
Enter allocation matrix:
1 0 0
0 1 0
0 0 1
Enter max matrix:
2 1 1
1 2 1
1 1 2
Enter available resources: 1 1 1
No deadlock detected.
```