S.B. JAIN INSTITUTE OF TECHNOLOGY MANAGEMENT & RESEARCH, NAGPUR

Practical 07

Aim: Develop a program to manage resource allocation for five processes (Google Drive, Firefox, Word Processor, Excel, and PowerPoint) using four types of resources (Printer, ROM, Hard Disk, and RAM). The program takes input for allocated, maximum, and available resources, calculates the current need of each process, and determines if a safe execution order exists using the Banker's Algorithm.

Name: Shraddha Agalawe

USN: EE24TCD002

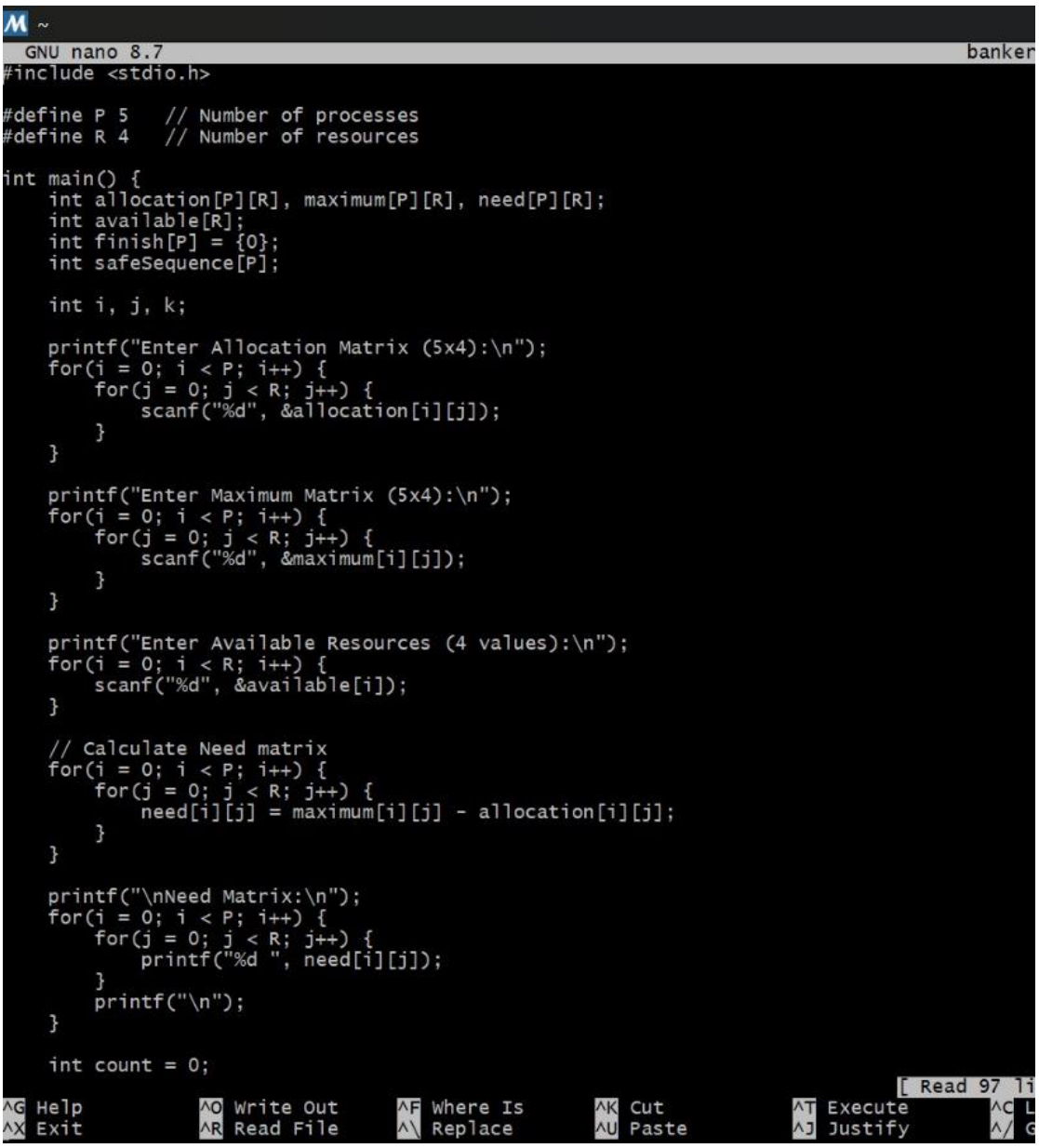
Semester / Year: 4th / 2nd

Academic Year: 2025-2026

Date of Performance:

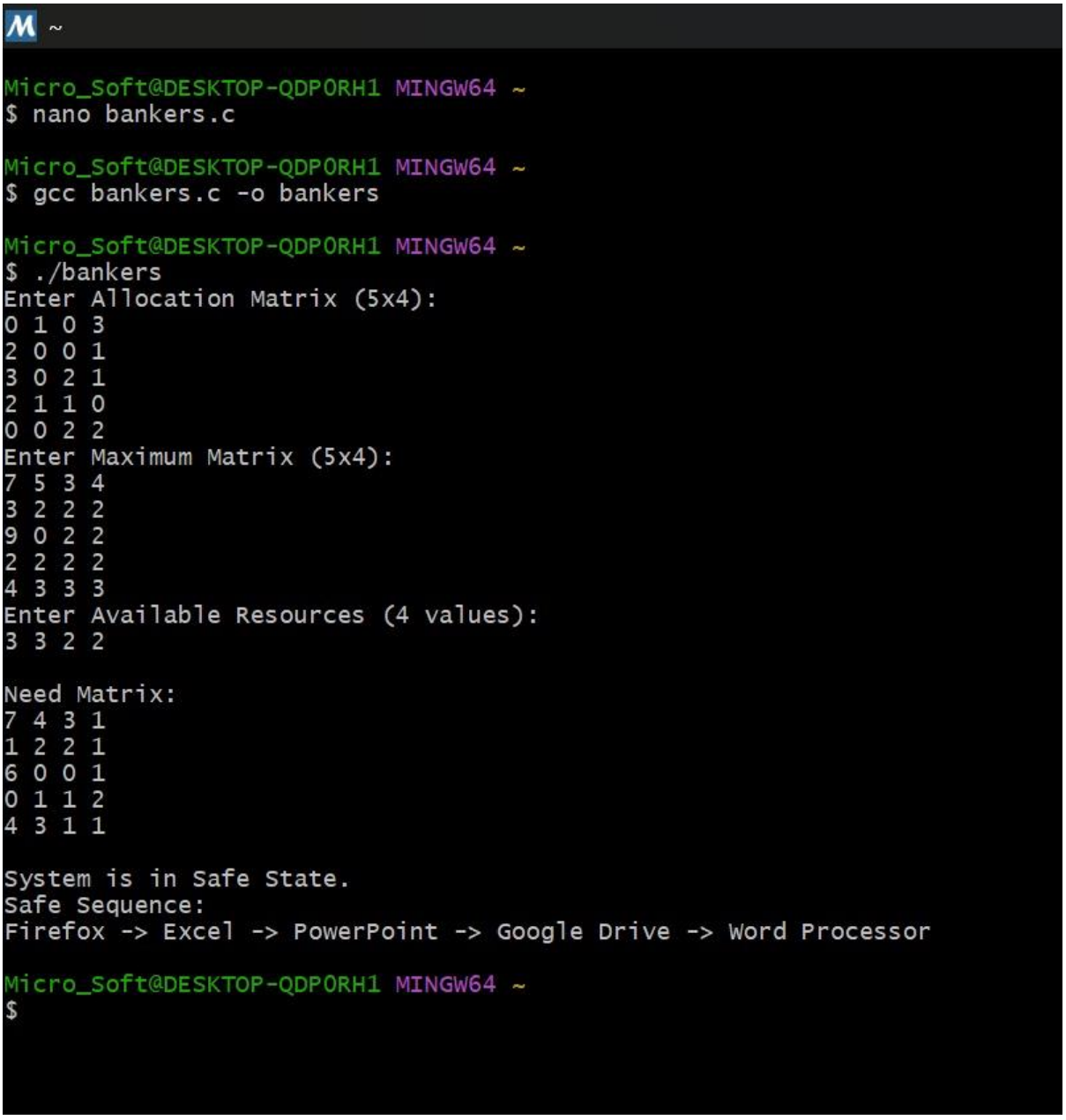
Date of Submission:

❖ CODE :



```
~
  GNU nano 8.7                                                    banker
#include <stdio.h>

#define P 5   // Number of processes
#define R 4   // Number of resources

int main() {
    int allocation[P][R], maximum[P][R], need[P][R];
    int available[R];
    int finish[P] = {0};
    int safeSequence[P];

    int i, j, k;

    printf("Enter Allocation Matrix (5x4):\n");
    for(i = 0; i < P; i++) {
        for(j = 0; j < R; j++) {
            scanf("%d", &allocation[i][j]);
        }
    }

    printf("Enter Maximum Matrix (5x4):\n");
    for(i = 0; i < P; i++) {
        for(j = 0; j < R; j++) {
            scanf("%d", &maximum[i][j]);
        }
    }

    printf("Enter Available Resources (4 values):\n");
    for(i = 0; i < R; i++) {
        scanf("%d", &available[i]);
    }

    // Calculate Need matrix
    for(i = 0; i < P; i++) {
        for(j = 0; j < R; j++) {
            need[i][j] = maximum[i][j] - allocation[i][j];
        }
    }

    printf("\nNeed Matrix:\n");
    for(i = 0; i < P; i++) {
        for(j = 0; j < R; j++) {
            printf("%d ", need[i][j]);
        }
        printf("\n");
    }

    int count = 0;
                                                          [ Read 97 li
^G Help          ^O Write Out     ^F Where Is      ^K Cut           ^T Execute       ^C L
^X Exit          ^R Read File     ^\ Replace       ^U Paste         ^J Justify       ^/ G
```

❖ OUTPUT :


~
Micro_Soft@DESKTOP-QDP0RH1 MINGW64 ~
$ nano bankers.c
Micro_Soft@DESKTOP-QDP0RH1 MINGW64 ~
$ gcc bankers.c -o bankers
Micro_Soft@DESKTOP-QDP0RH1 MINGW64 ~
$ ./bankers
Enter Allocation Matrix (5x4):
0 1 0 3
2 0 0 1
3 0 2 1
2 1 1 0
0 0 2 2
Enter Maximum Matrix (5x4):
7 5 3 4
3 2 2 2
9 0 2 2
2 2 2 2
4 3 3 3
Enter Available Resources (4 values):
3 3 2 2
Need Matrix:
7 4 3 1
1 2 2 1
6 0 0 1
0 1 1 2
4 3 1 1
System is in Safe State.
Safe Sequence:
Firefox -> Excel -> PowerPoint -> Google Drive -> Word Processor
Micro_Soft@DESKTOP-QDP0RH1 MINGW64 ~
$