Lab Report 8

Fall 2024

Course Title: Structured Programming Lab

Course Code: CSE 1202 (Fall 2024)

Submitted by: **Student**

Name and ID

Md. Tazminur Rahman Tanim (242014124)

Department of CSE

University of Liberal Arts Bangladesh (ULAB)

1. Write a program to read two matrices from user into two different 2D array and multiply these
two matrixes and finally display the result.

Answer :

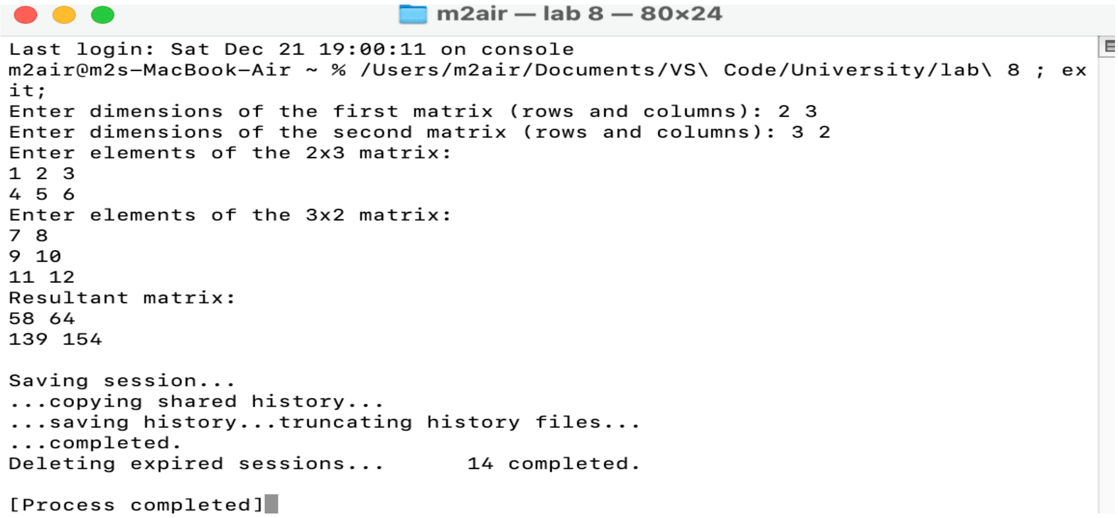
Algorithm:

- Start.
- Input the dimensions of the two matrices A and B such that the number of columns in A equals the number of rows in B.
- Input the elements of matrix A and B.
- Initialize a result matrix C with all elements set to 0.
- Perform matrix multiplication.
- Display the result matrix C.
- End.

Code

```
#include <stdio.h>

void readMatrix(int rows, int cols, int matrix[rows][cols]) {
    printf("Enter elements of the %dx%d matrix:\n", rows, cols);
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            scanf("%d", &matrix[i][j]);
        }
    }
}

void multiplyMatrices(int rowsA, int colsA, int matrixA[rowsA][colsA],
                      int rowsB, int colsB, int matrixB[rowsB][colsB],
                      int result[rowsA][colsB]) {
    for (int i = 0; i < rowsA; i++) {
        for (int j = 0; j < colsB; j++) {
            result[i][j] = 0;
            for (int k = 0; k < colsA; k++) {
                result[i][j] += matrixA[i][k] * matrixB[k][j];
            }
        }
    }
}

void displayMatrix(int rows, int cols, int matrix[rows][cols]) {
    printf("Resultant matrix:\n");
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            printf("%d ", matrix[i][j]);
        }
        printf("\n");
    }
}

int main() {
    int rowsA, colsA, rowsB, colsB;

    printf("Enter dimensions of the first matrix (rows and columns): ");
    scanf("%d %d", &rowsA, &colsA);

    printf("Enter dimensions of the second matrix (rows and columns): ");
    scanf("%d %d", &rowsB, &colsB);

    if (colsA != rowsB) {
        printf("Matrix multiplication is not possible.\n");
        return 1;
    }

    int matrixA[rowsA][colsA], matrixB[rowsB][colsB], result[rowsA][colsB];

    readMatrix(rowsA, colsA, matrixA);
    readMatrix(rowsB, colsB, matrixB);

    multiplyMatrices(rowsA, colsA, matrixA, rowsB, colsB, matrixB, result);

    displayMatrix(rowsA, colsB, result);

    return 0;
}
```

Output Result :



```
Last login: Sat Dec 21 19:00:11 on console
m2air@m2s-MacBook-Air ~ % /Users/m2air/Documents/VS\ Code/University/lab\ 8 ; ex
it;
Enter dimensions of the first matrix (rows and columns): 2 3
Enter dimensions of the second matrix (rows and columns): 3 2
Enter elements of the 2x3 matrix:
1 2 3
4 5 6
Enter elements of the 3x2 matrix:
7 8
9 10
11 12
Resultant matrix:
58 64
139 154

Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.
Deleting expired sessions...      14 completed.

[Process completed]
```

2. Write a program to display the numbers divisible by 4 or 7 from a 2D array of integers and find the sum of those numbers as well.

Answer :

Algorithm:

- Start.
- Input the dimensions of the 2D array.
- Input the elements of the array.
- Traverse through the array and check each element:

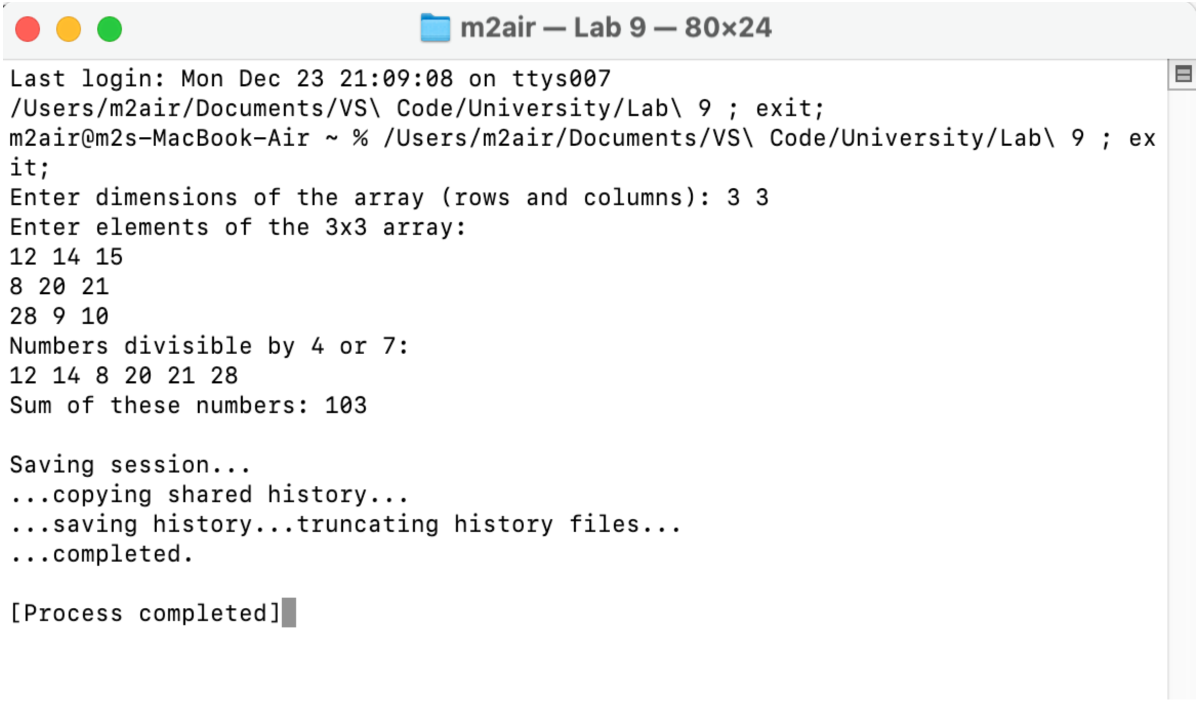
  1.If the element is divisible by 4 or 7, add it to the sum and display it.

- Display the total sum of the divisible numbers.
- End.

Code

```
#include <stdio.h>

void readMatrix(int rows, int cols, int matrix[rows][cols]) {
    printf("Enter elements of the %dx%d array:\n", rows, cols);
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            scanf("%d", &matrix[i][j]);
        }
    }
}

void findAndSumDivisible(int rows, int cols, int matrix[rows][cols]) {
    int sum = 0;
    printf("Numbers divisible by 4 or 7:\n");
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            if (matrix[i][j] % 4 == 0 || matrix[i][j] % 7 == 0) {
                printf("%d ", matrix[i][j]);
                sum += matrix[i][j];
            }
        }
    }
    printf("\nSum of these numbers: %d\n", sum);
}

int main() {
    int rows, cols;

    printf("Enter dimensions of the array (rows and columns): ");
    scanf("%d %d", &rows, &cols);

    int matrix[rows][cols];

    readMatrix(rows, cols, matrix);

    findAndSumDivisible(rows, cols, matrix);

    return 0;
}
```

Output Result :



```
Last login: Mon Dec 23 21:09:08 on ttys007
/Users/m2air/Documents/VS\ Code/University/Lab\ 9 ; exit;
m2air@m2s-MacBook-Air ~ % /Users/m2air/Documents/VS\ Code/University/Lab\ 9 ; ex
it;
Enter dimensions of the array (rows and columns): 3 3
Enter elements of the 3x3 array:
12 14 15
8 20 21
28 9 10
Numbers divisible by 4 or 7:
12 14 8 20 21 28
Sum of these numbers: 103

Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.

[Process completed]
```

Discussion : Matrix multiplication involves computing a new matrix by taking the dot product of rows and columns from two input matrices, requiring matching dimensions. It’s essential in areas like graphics and machine learning.The second task identifies numbers divisible by 4 or 7 in a 2D array and calculates their sum, showcasing modular arithmetic and array traversal. Both problems highlight key programming concepts in mathematics and data processing.