KARTHIK KRISHNAN
S2 CSE B
ROLL: 45

LAB - 8

C PROGRAMMING

28-04-2024

Menu Driven Program

QN) Write a menu driven program for performing matrix addition, multiplication and finding the transpose. Use functions to (i) read a matrix, (ii) find the sum of two matrices, (iii) find the product of two matrices, (i) find the transpose of a matrix and (v) display a matrix.

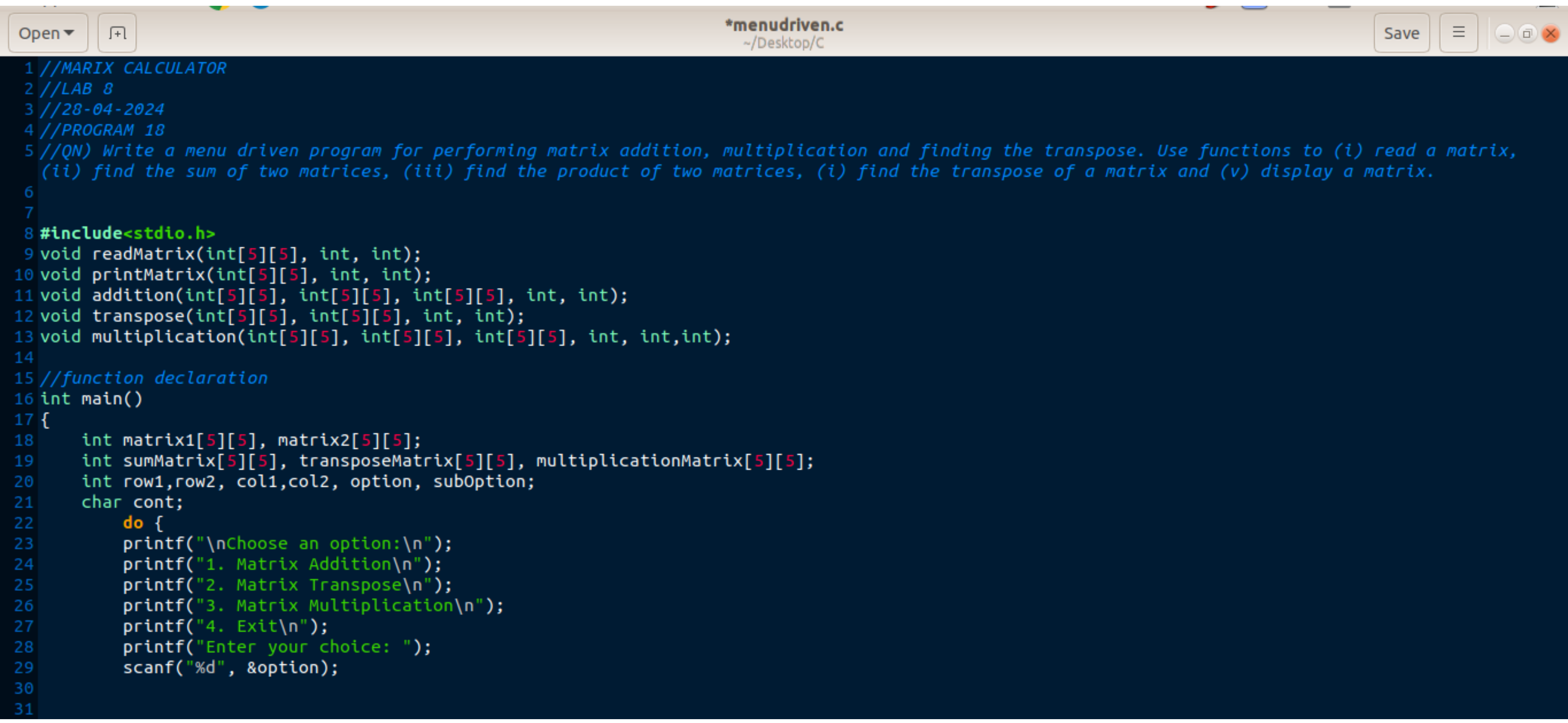
Ans:

Program Code

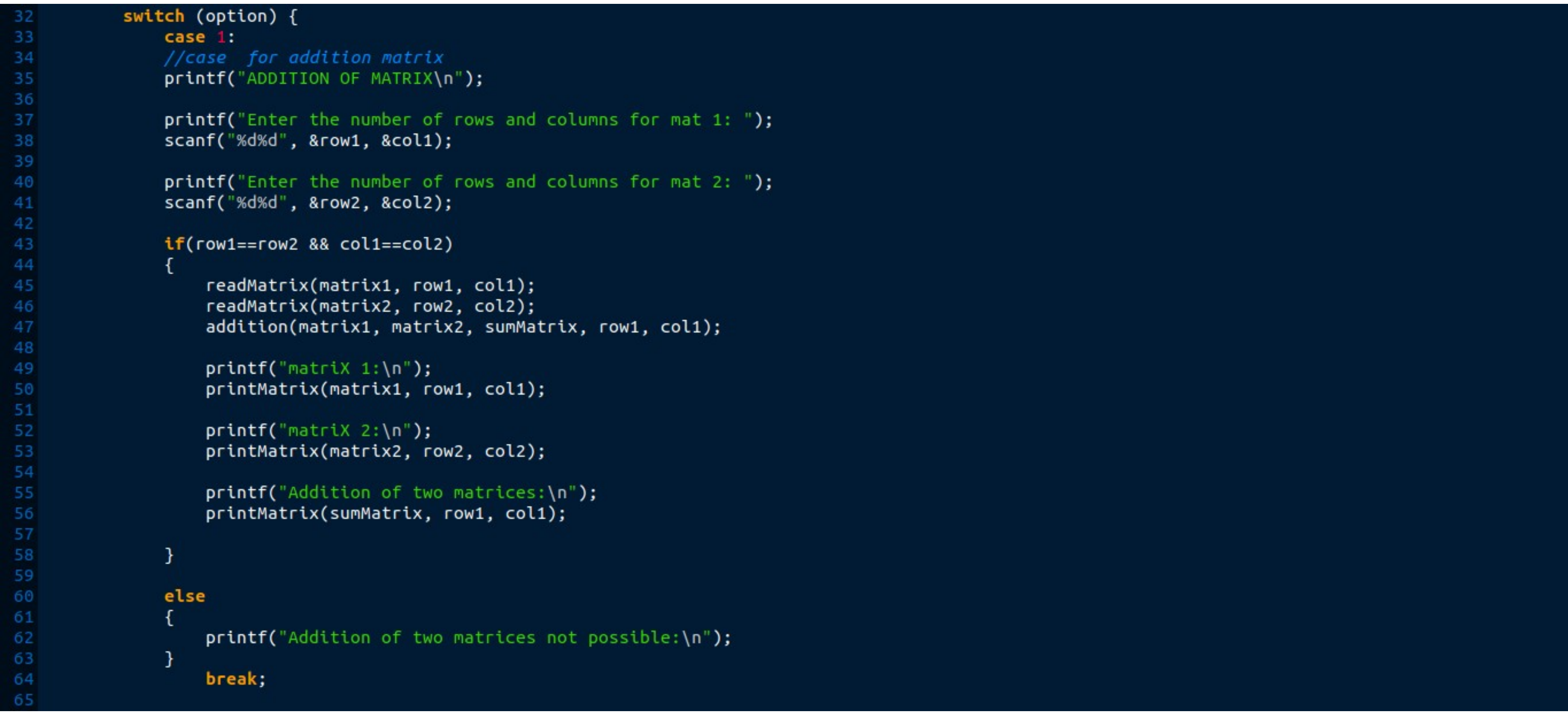
```
#include<stdio.h>
void readMatrix(int [5] [5], int, int);
void printMatrix(int [5] [5], int, int);
void addition(int [5] [5], int [5] [5], int [5] [5], int, int);
void transpose(int [5] [5], int [5] [5], int, int);
void multiplication(int[5] [5], int[5] [5], int [5] [5], int, int,int);

//function declaration
int main()
{
   int matrix1[5] [5], matrix2[5] [5];
   int sumMatrix[5] [5], transposeMatrix[5] [5], multiplicationMatrix[5] [5];
   int row1,row2, col1,col2, option, subOption;
   char cont;
      do {
      printf("\nChoose an option:\n");
printf("1. Matrix Addition\n");
printf("2. Matrix Transpose\n");
printf("3. Matrix Multiplication\n");
printf("4. Exit\n");
printf("Enter your choice:  ");
scanf("%d", &option);


switch (option) {
    case 1:
    //case  for addition matrix
    printf("ADDITION OF MATRIX\n");

    printf("Enter the number of rows and columns for mat 1: ");
    scanf("%d%d", &row1, &col1);

    printf("Enter the number of rows and columns for mat 2: ");
    scanf("%d%d", &row2, &col2);

    if(row1==row2 && col1==col2)
    {
        readMatrix(matrix1, row1, col1);
        readMatrix(matrix2, row2, col2);
        addition(matrix1, matrix2, sumMatrix, row1, col1);

        printf("matriX 1:\n");
        printMatrix(matrix1, row1, col1);

        printf("matriX 2:\n");
        printMatrix(matrix2, row2, col2);

        printf("Addition of two matrices:\n");
        printMatrix(sumMatrix, row1, col1);
}

else
{
    printf("Addition of two matrices not possible:\n");
}
    break;

    case 2:
    //case for transpose matrix
    printf("TRANSPOSE OF MATRIX\n");
    printf("Enter the number of rows and columns for the matrix: ");
    scanf("%d %d", &row1, &col1);

     // Read the matrix from the user
    readMatrix(matrix1, row1, col1);

     // Transpose the matrix
    transpose(matrix1, transposeMatrix, row1, col1);

     // Print the original matrix
    printf("Original Matrix: \n");
    printMatrix(matrix1, row1, col1);

     // Print the transposed matrix
    printf("Transposed Matrix: \n");
    printMatrix(transposeMatrix, col1, row1);

    break;

    case 3:
    //case for multiplication matrix
```

```c
        printf("MULTIPLICATION OF MATRIX\n");
        printf("Enter no of rows and columns in matrix1: ");
        scanf("%d%d",&row1,&col1);
        printf("Enter no of rows and columns in matrix2: ");
        scanf("%d%d",&row2,&col2);
        if (col1!=row2)
        {
            printf("Multiplication not possible\n");
            break;

        }

        printf("MATRIX 1");
        printf("\n");

        readMatrix(matrix1,row1,row2);
        printf("MATRIX 2");
        printf("\n");

        readMatrix(matrix2,row2,col2);
        multiplication(matrix1, matrix2, multiplicationMatrix, row1, col2,
col1);

        printf("Multiplication of two matrices: \n");
        printMatrix(multiplicationMatrix,row1,col2);
        break;

        case 4:
         // Case for exiting the program
        printf("Exiting the program.\n");
        break;

        default:
                printf("Invalid option!\n");

        }
        printf("Do you want to continue? (y/n): ");
        scanf(" %c", &cont);

        }
        while (cont == 'y' || cont == 'Y');

        printf("Exiting...\n");
        return 0;
}



//Function Definition
void readMatrix(int matrix[5][5], int row, int col) {
    printf("Enter the elements:\n");
    for (int i = 0; i < row; i++)
    {
        for (int j = 0; j < col; j++)
        {
            scanf("%d", &matrix[i][j]);
        }
    }
}

void printMatrix(int matrix[5][5], int row, int col) {
    for (int i = 0; i < row; i++) {
        for (int j = 0; j < col; j++) {
            printf("%d\t", matrix[i][j]);
        }
        printf("\n");
    }
```

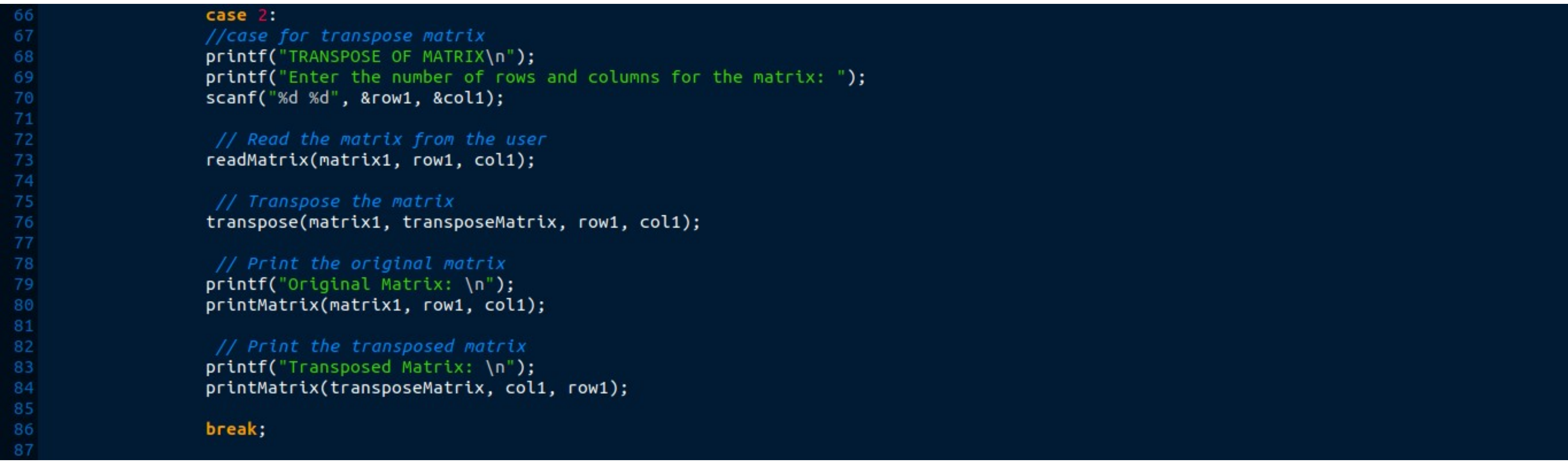
```
}

//Addition - Function definiton
void addition(int matrix1[5] [5], int matrix2[5] [5], int sumMatrix[5] [5], int
row, int col) {
    for (int i = 0; i < row; i++) {
        for (int j = 0; j < col; j++) {
            sumMatrix[i] [j] = matrix1[i] [j] + matrix2[i] [j];
        }
    }
}
//Transpose - Function definition
void transpose(int matrix[5] [5], int transMatrix[5] [5], int row, int col) {
    for (int i = 0; i < row; i++) {
        for (int j = 0; j < col; j++) {
            transMatrix[i] [j] = matrix[j][i];
        }
    }
}

//Multiplication - Function definition
void multiplication(int matrix1[5] [5], int matrix2[5] [5], int
multiplicationMatrix[5] [5], int row1, int col2, int col1){
    for(int i=0; i<row1 ; i++){
        for(int j=0; j<col2;j++){
            multiplicationMatrix[i] [j] = 0;
            for(int k=0; k<col1;k++){
                multiplicationMatrix[i] [j] += matrix1[i] [k] * matrix2[k] [j];
            }
        }
    }
}
```



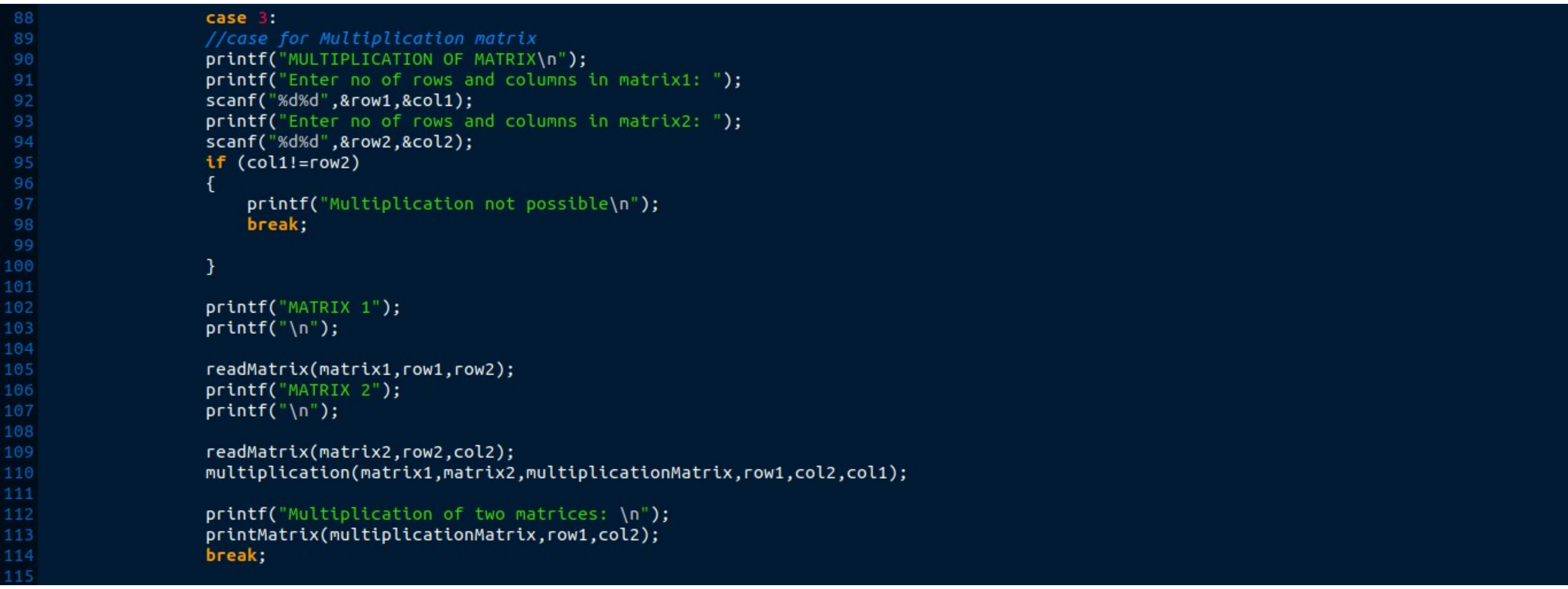
```c
//MARIX CALCULATOR
//LAB 8
//28-04-2024
//PROGRAM 18
//QN) Write a menu driven program for performing matrix addition, multiplication and finding the transpose. Use functions to (i) read a matrix, (ii) find the sum of two matrices, (iii) find the product of two matrices, (i) find the transpose of a matrix and (v) display a matrix.


#include<stdio.h>
void readMatrix(int[5][5], int, int);
void printMatrix(int[5][5], int, int);
void addition(int[5][5], int[5][5], int[5][5], int, int);
void transpose(int[5][5], int[5][5], int, int);
void multiplication(int[5][5], int[5][5], int[5][5], int, int,int);

//function declaration
int main()
{
    int matrix1[5][5], matrix2[5][5];
    int sumMatrix[5][5], transposeMatrix[5][5], multiplicationMatrix[5][5];
    int row1,row2, col1,col2, option, subOption;
    char cont;
        do {
        printf("\nChoose an option:\n");
        printf("1. Matrix Addition\n");
        printf("2. Matrix Transpose\n");
        printf("3. Matrix Multiplication\n");
        printf("4. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &option);


```



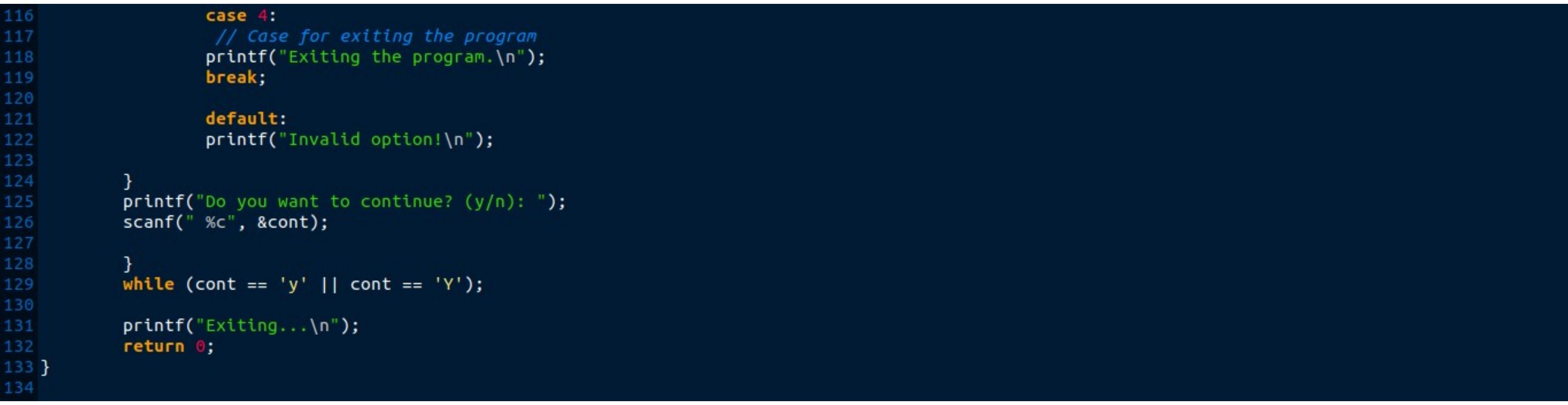
```c
        switch (option) {
            case 1:
            //case  for addition matrix
            printf("ADDITION OF MATRIX\n");

            printf("Enter the number of rows and columns for mat 1: ");
            scanf("%d%d", &row1, &col1);

            printf("Enter the number of rows and columns for mat 2: ");
            scanf("%d%d", &row2, &col2);

            if(row1==row2 && col1==col2)
            {
                readMatrix(matrix1, row1, col1);
                readMatrix(matrix2, row2, col2);
                addition(matrix1, matrix2, sumMatrix, row1, col1);

                printf("matriX 1:\n");
                printMatrix(matrix1, row1, col1);

                printf("matriX 2:\n");
                printMatrix(matrix2, row2, col2);

                printf("Addition of two matrices:\n");
                printMatrix(sumMatrix, row1, col1);

            }

            else
            {
                printf("Addition of two matrices not possible:\n");
            }
                break;

```



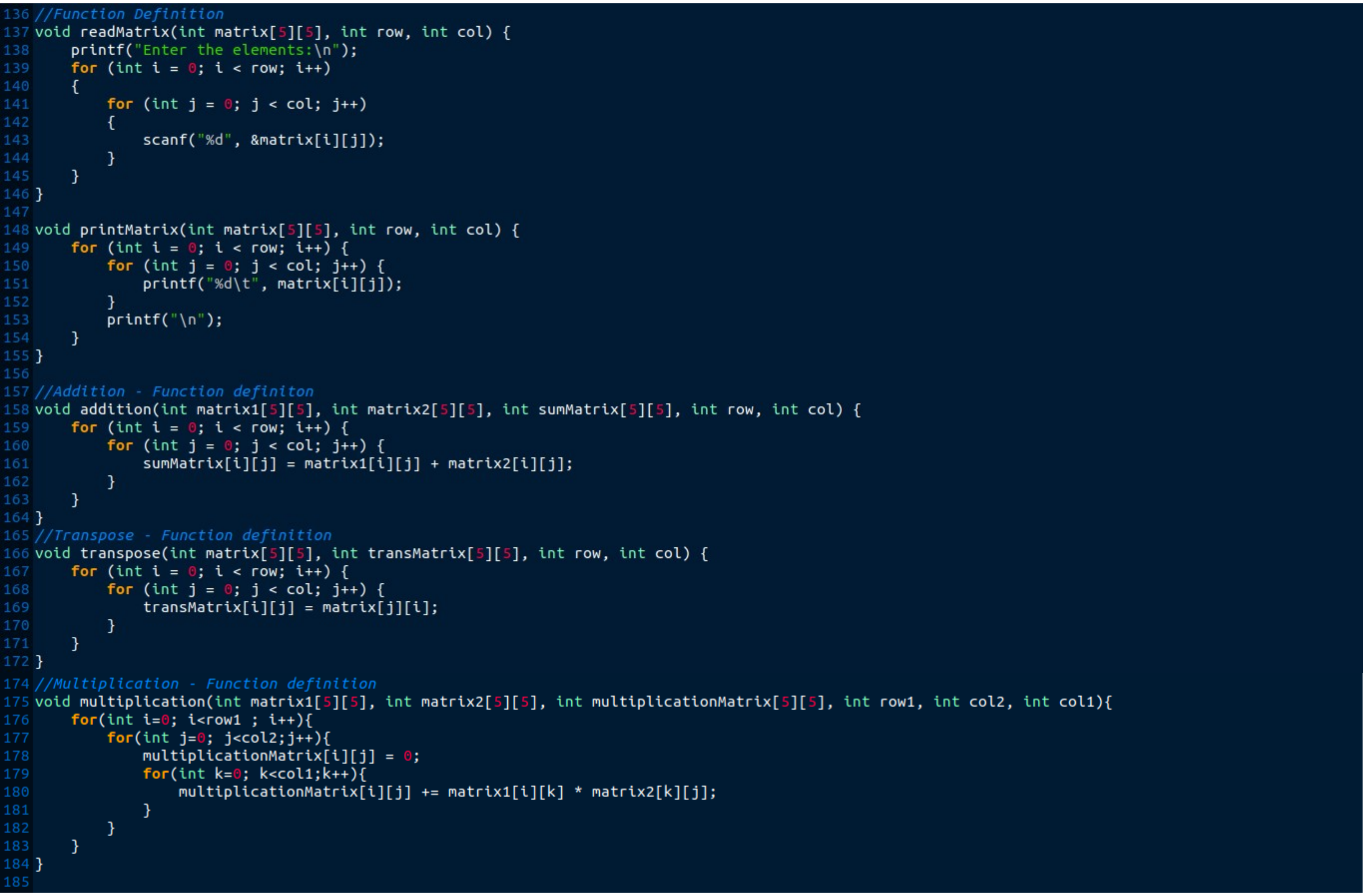
```c
                case 2:
                //case for transpose matrix
                printf("TRANSPOSE OF MATRIX\n");
                printf("Enter the number of rows and columns for the matrix: ");
                scanf("%d %d", &row1, &col1);

                 // Read the matrix from the user
                readMatrix(matrix1, row1, col1);

                 // Transpose the matrix
                transpose(matrix1, transposeMatrix, row1, col1);

                 // Print the original matrix
                printf("Original Matrix: \n");
                printMatrix(matrix1, row1, col1);

                 // Print the transposed matrix
                printf("Transposed Matrix: \n");
                printMatrix(transposeMatrix, col1, row1);

                break;

```



```c
                case 3:
                //case for Multiplication matrix
                printf("MULTIPLICATION OF MATRIX\n");
                printf("Enter no of rows and columns in matrix1: ");
                scanf("%d%d",&row1,&col1);
                printf("Enter no of rows and columns in matrix2: ");
                scanf("%d%d",&row2,&col2);
                if (col1!=row2)
                {
                    printf("Multiplication not possible\n");
                    break;

                }

                printf("MATRIX 1");
                printf("\n");

                readMatrix(matrix1,row1,row2);
                printf("MATRIX 2");
                printf("\n");

                readMatrix(matrix2,row2,col2);
                multiplication(matrix1,matrix2,multiplicationMatrix,row1,col2,col1);

                printf("Multiplication of two matrices: \n");
                printMatrix(multiplicationMatrix,row1,col2);
                break;

```



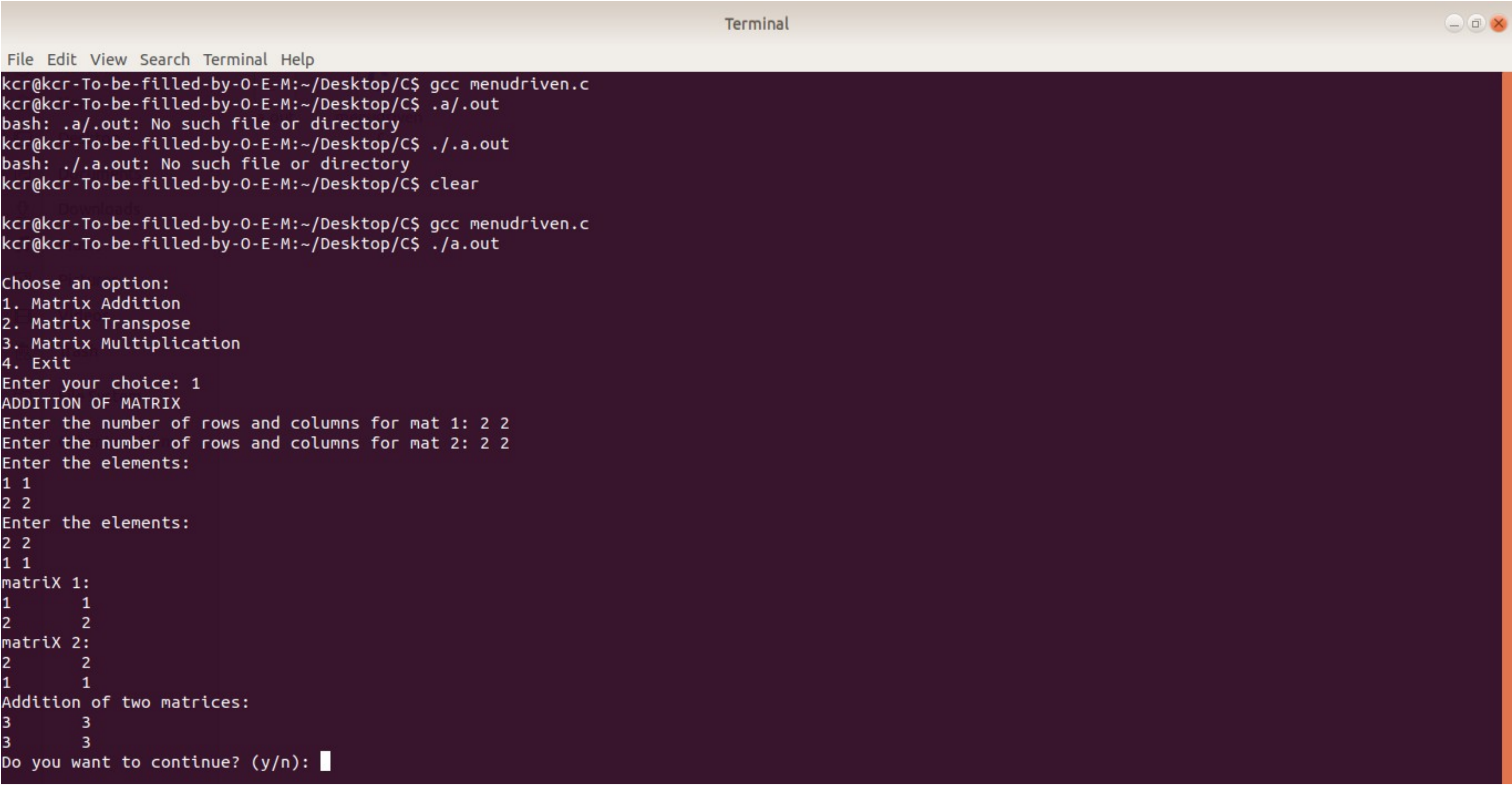
```c
                case 4:
                 // Case for exiting the program
                printf("Exiting the program.\n");
                break;

                default:
                printf("Invalid option!\n");

        }
        printf("Do you want to continue? (y/n): ");
        scanf(" %c", &cont);

        }
        while (cont == 'y' || cont == 'Y');

        printf("Exiting...\n");
        return 0;
}

```



```c
//Function Definition
void readMatrix(int matrix[5][5], int row, int col) {
    printf("Enter the elements:\n");
    for (int i = 0; i < row; i++)
    {
        for (int j = 0; j < col; j++)
        {
            scanf("%d", &matrix[i][j]);
        }
    }
}

void printMatrix(int matrix[5][5], int row, int col) {
    for (int i = 0; i < row; i++) {
        for (int j = 0; j < col; j++) {
            printf("%d\t", matrix[i][j]);
        }
        printf("\n");
    }
}

//Addition - Function definiton
void addition(int matrix1[5][5], int matrix2[5][5], int sumMatrix[5][5], int row, int col) {
    for (int i = 0; i < row; i++) {
        for (int j = 0; j < col; j++) {
            sumMatrix[i][j] = matrix1[i][j] + matrix2[i][j];
        }
    }
}
//Transpose - Function definition
void transpose(int matrix[5][5], int transMatrix[5][5], int row, int col) {
    for (int i = 0; i < row; i++) {
        for (int j = 0; j < col; j++) {
            transMatrix[i][j] = matrix[j][i];
        }
    }
}
//Multiplication - Function definition
void multiplication(int matrix1[5][5], int matrix2[5][5], int multiplicationMatrix[5][5], int row1, int col2, int col1){
    for(int i=0; i<row1 ; i++){
        for(int j=0; j<col2;j++){
            multiplicationMatrix[i][j] = 0;
            for(int k=0; k<col1;k++){
                multiplicationMatrix[i][j] += matrix1[i][k] * matrix2[k][j];
            }
        }
    }
}

```

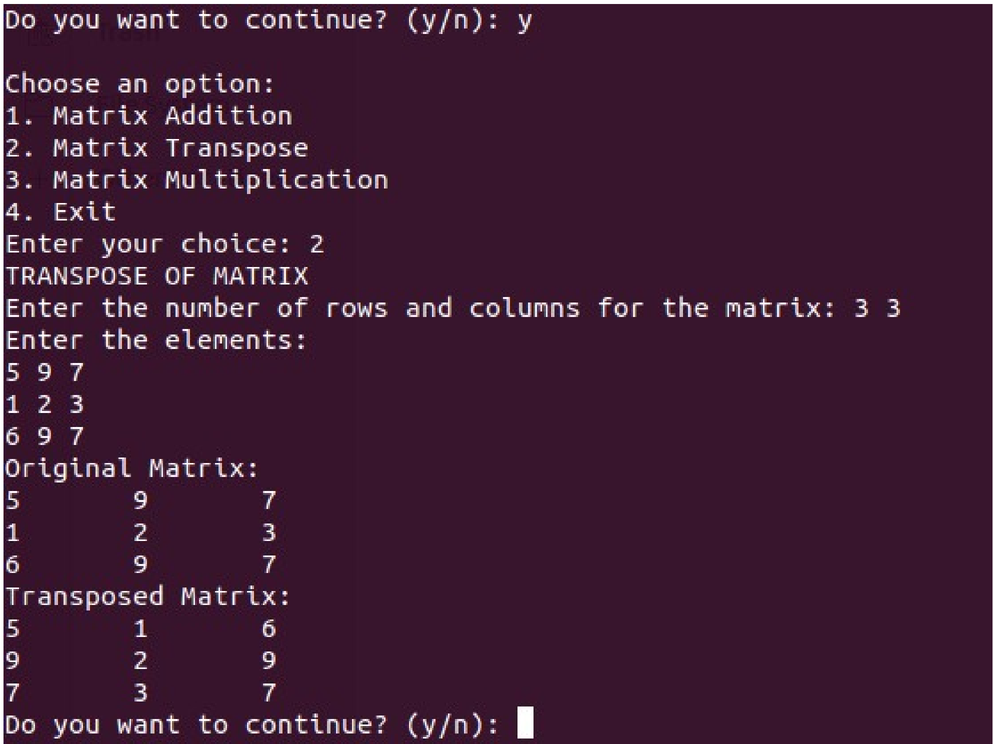
OUTPUT

(i) ADDITION



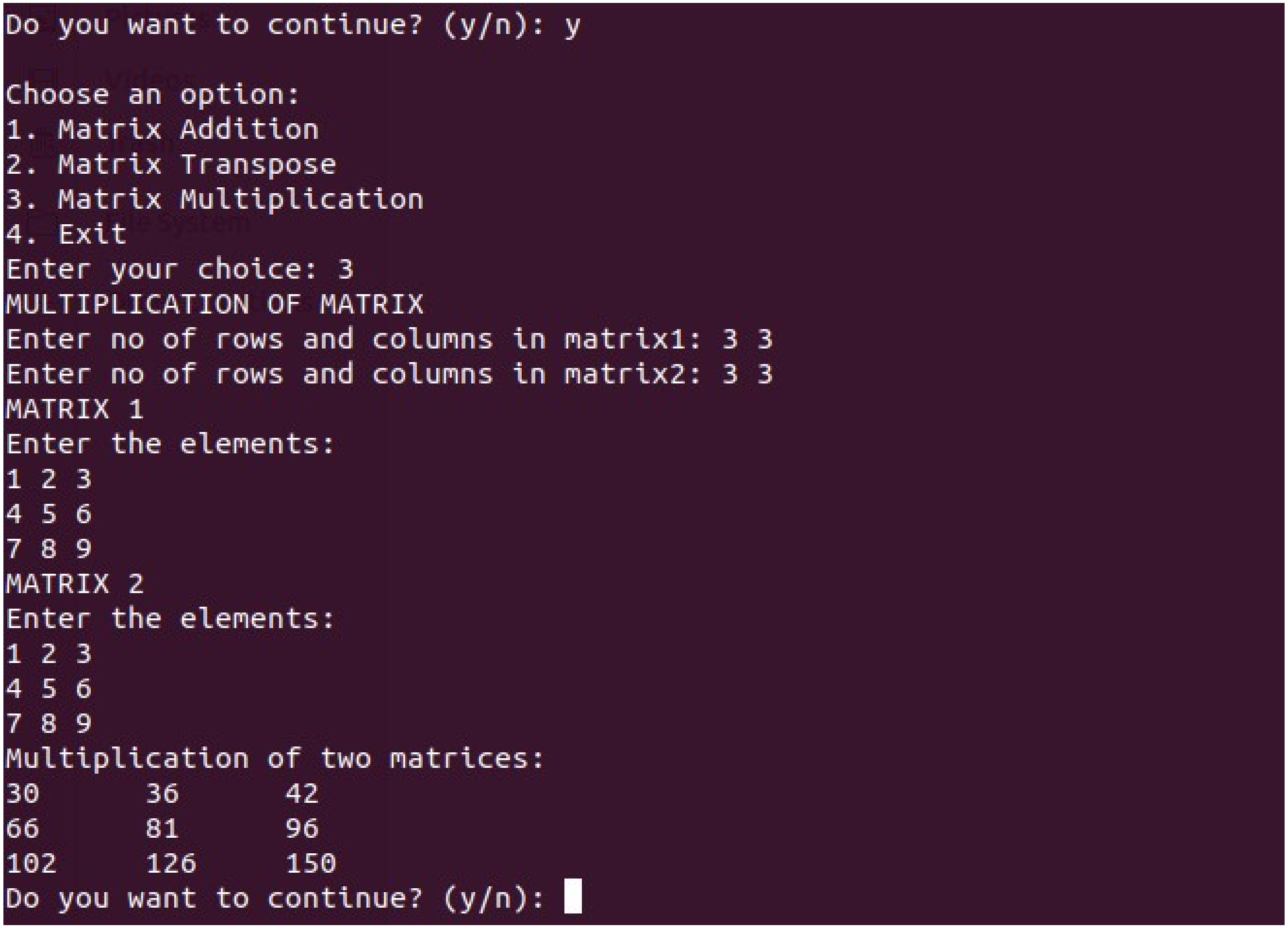
```
kcr@kcr-To-be-filled-by-O-E-M:~/Desktop/C$ gcc menudriven.c
kcr@kcr-To-be-filled-by-O-E-M:~/Desktop/C$ .a/.out
bash: .a/.out: No such file or directory
kcr@kcr-To-be-filled-by-O-E-M:~/Desktop/C$ ./.a.out
bash: ./.a.out: No such file or directory
kcr@kcr-To-be-filled-by-O-E-M:~/Desktop/C$ clear

kcr@kcr-To-be-filled-by-O-E-M:~/Desktop/C$ gcc menudriven.c
kcr@kcr-To-be-filled-by-O-E-M:~/Desktop/C$ ./a.out

Choose an option:
1. Matrix Addition
2. Matrix Transpose
3. Matrix Multiplication
4. Exit
Enter your choice: 1
ADDITION OF MATRIX
Enter the number of rows and columns for mat 1: 2 2
Enter the number of rows and columns for mat 2: 2 2
Enter the elements:
1 1
2 2
Enter the elements:
2 2
1 1
matriX 1:
1       1
2       2
matriX 2:
2       2
1       1
Addition of two matrices:
3       3
3       3
Do you want to continue? (y/n):
```

(ii) TRANSPOSE



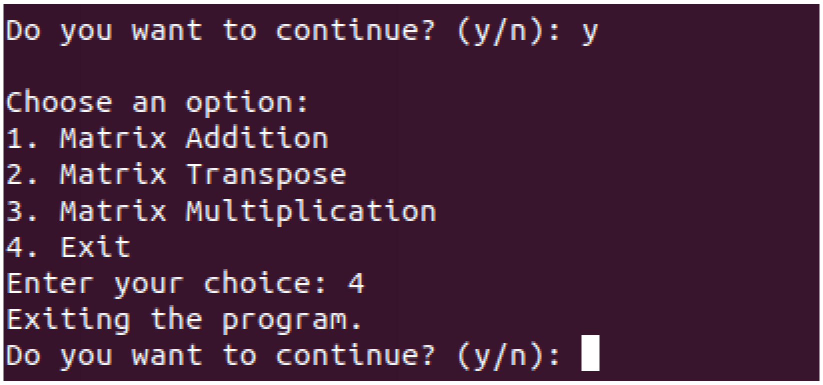
```
Do you want to continue? (y/n): y

Choose an option:
1. Matrix Addition
2. Matrix Transpose
3. Matrix Multiplication
4. Exit
Enter your choice: 2
TRANSPOSE OF MATRIX
Enter the number of rows and columns for the matrix: 3 3
Enter the elements:
5 9 7
1 2 3
6 9 7
Original Matrix:
5       9       7
1       2       3
6       9       7
Transposed Matrix:
5       1       6
9       2       9
7       3       7
Do you want to continue? (y/n):
```

(iii) MULTIPLICATION



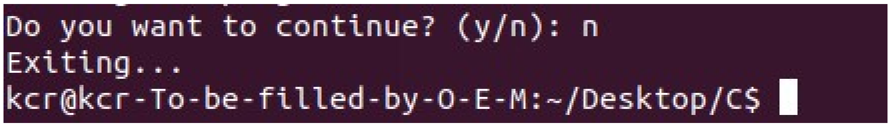
```
Do you want to continue? (y/n): y

Choose an option:
1. Matrix Addition
2. Matrix Transpose
3. Matrix Multiplication
4. Exit
Enter your choice: 3
MULTIPLICATION OF MATRIX
Enter no of rows and columns in matrix1: 3 3
Enter no of rows and columns in matrix2: 3 3
MATRIX 1
Enter the elements:
1 2 3
4 5 6
7 8 9
MATRIX 2
Enter the elements:
1 2 3
4 5 6
7 8 9
Multiplication of two matrices:
30      36      42
66      81      96
102     126     150
Do you want to continue? (y/n): 
```

(EXIST)



```
Do you want to continue? (y/n): y

Choose an option:
1. Matrix Addition
2. Matrix Transpose
3. Matrix Multiplication
4. Exit
Enter your choice: 4
Exiting the program.
Do you want to continue? (y/n): 
```

Yes or no



```
Do you want to continue? (y/n): n
Exiting...
kcr@kcr-To-be-filled-by-O-E-M:~/Desktop/C$ 
```