

**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, (iv) 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("TRANPOSE 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
```

```

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];  
            }  
        }  
    }  
}
```

```
1 //MARIX CALCULATOR
2 //LAB 8
3 //28-04-2024
4 //PROGRAM 18
5 //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.
6
7
8 #include<stdio.h>
9 void readMatrix(int[5][5], int, int);
10 void printMatrix(int[5][5], int, int);
11 void addition(int[5][5], int[5][5], int[5][5], int, int);
12 void transpose(int[5][5], int[5][5], int, int);
13 void multiplication(int[5][5], int[5][5], int[5][5], int, int,int);
14
15 //function declaration
16 int main()
17 {
18     int matrix1[5][5], matrix2[5][5];
19     int sumMatrix[5][5], transposeMatrix[5][5], multiplicationMatrix[5][5];
20     int row1,row2, col1,col2, option, subOption;
21     char cont;
22     do {
23         printf("\nChoose an option:\n");
24         printf("1. Matrix Addition\n");
25         printf("2. Matrix Transpose\n");
26         printf("3. Matrix Multiplication\n");
27         printf("4. Exit\n");
28         printf("Enter your choice: ");
29         scanf("%d", &option);
30
31
```

```
32     switch (option) {
33         case 1:
34             //case for addition matrix
35             printf("ADDITION OF MATRIX\n");
36
37             printf("Enter the number of rows and columns for mat 1: ");
38             scanf("%d%d", &row1, &col1);
39
40             printf("Enter the number of rows and columns for mat 2: ");
41             scanf("%d%d", &row2, &col2);
42
43             if(row1==row2 && col1==col2)
44             {
45                 readMatrix(matrix1, row1, col1);
46                 readMatrix(matrix2, row2, col2);
47                 addition(matrix1, matrix2, sumMatrix, row1, col1);
48
49                 printf("matrix 1:\n");
50                 printMatrix(matrix1, row1, col1);
51
52                 printf("matrix 2:\n");
53                 printMatrix(matrix2, row2, col2);
54
55                 printf("Addition of two matrices:\n");
56                 printMatrix(sumMatrix, row1, col1);
57
58             }
59
60         else
61         {
62             printf("Addition of two matrices not possible:\n");
63         }
64         break;
65
```

```

66         case 2:
67             //case for transpose matrix
68             printf("TRANSPOSE OF MATRIX\n");
69             printf("Enter the number of rows and columns for the matrix: ");
70             scanf("%d %d", &row1, &col1);
71
72             // Read the matrix from the user
73             readMatrix(matrix1, row1, col1);
74
75             // Transpose the matrix
76             transpose(matrix1, transposeMatrix, row1, col1);
77
78             // Print the original matrix
79             printf("Original Matrix: \n");
80             printMatrix(matrix1, row1, col1);
81
82             // Print the transposed matrix
83             printf("Transposed Matrix: \n");
84             printMatrix(transposeMatrix, col1, row1);
85
86             break;
87
88         case 3:
89             //case for Multiplication matrix
90             printf("MULTIPLICATION OF MATRIX\n");
91             printf("Enter no of rows and columns in matrix1: ");
92             scanf("%d%d",&row1,&col1);
93             printf("Enter no of rows and columns in matrix2: ");
94             scanf("%d%d",&row2,&col2);
95             if (col1!=row2)
96             {
97                 printf("Multiplication not possible\n");
98                 break;
99             }
100
101             printf("MATRIX 1");
102             printf("\n");
103
104             readMatrix(matrix1,row1,row2);
105             printf("MATRIX 2");
106             printf("\n");
107
108             readMatrix(matrix2,row2,col2);
109             multiplication(matrix1,matrix2,multiplicationMatrix,row1,col2,col1);
110
111             printf("Multiplication of two matrices: \n");
112             printMatrix(multiplicationMatrix,row1,col2);
113             break;
114
115
116         case 4:
117             // Case for exiting the program
118             printf("Exiting the program.\n");
119             break;
120
121         default:
122             printf("Invalid option!\n");
123
124     }
125     printf("Do you want to continue? (y/n): ");
126     scanf(" %c", &cont);
127
128     }
129     while (cont == 'y' || cont == 'Y');
130
131     printf("Exiting...\n");
132     return 0;
133 }
134

```



```

136 //Function Definition
137 void readMatrix(int matrix[5][5], int row, int col) {
138     printf("Enter the elements:\n");
139     for (int i = 0; i < row; i++)
140     {
141         for (int j = 0; j < col; j++)
142         {
143             scanf("%d", &matrix[i][j]);
144         }
145     }
146 }
147
148 void printMatrix(int matrix[5][5], int row, int col) {
149     for (int i = 0; i < row; i++) {
150         for (int j = 0; j < col; j++) {
151             printf("%d\t", matrix[i][j]);
152         }
153         printf("\n");
154     }
155 }
156
157 //Addition - Function definition
158 void addition(int matrix1[5][5], int matrix2[5][5], int sumMatrix[5][5], int row, int col) {
159     for (int i = 0; i < row; i++) {
160         for (int j = 0; j < col; j++) {
161             sumMatrix[i][j] = matrix1[i][j] + matrix2[i][j];
162         }
163     }
164 }
165 //Transpose - Function definition
166 void transpose(int matrix[5][5], int transMatrix[5][5], int row, int col) {
167     for (int i = 0; i < row; i++) {
168         for (int j = 0; j < col; j++) {
169             transMatrix[i][j] = matrix[j][i];
170         }
171     }
172 }
173
174 //Multiplication - Function definition
175 void multiplication(int matrix1[5][5], int matrix2[5][5], int multiplicationMatrix[5][5], int row1, int col2, int col1){
176     for(int i=0; i<row1 ; i++){
177         for(int j=0; j<col2;j++){
178             multiplicationMatrix[i][j] = 0;
179             for(int k=0; k<col1;k++){
180                 multiplicationMatrix[i][j] += matrix1[i][k] * matrix2[k][j];
181             }
182         }
183     }
184 }
185

```

# OUTPUT

## (i) ADDITION

```
Terminal
File Edit View Search Terminal Help
kcr@kcr-To-be-filled-by-0-E-M:~/Desktop/C$ gcc menudriven.c
kcr@kcr-To-be-filled-by-0-E-M:~/Desktop/C$ ./a.out
bash: ./a.out: No such file or directory
kcr@kcr-To-be-filled-by-0-E-M:~/Desktop/C$ ./a.out
bash: ./a.out: No such file or directory
kcr@kcr-To-be-filled-by-0-E-M:~/Desktop/C$ clear

kcr@kcr-To-be-filled-by-0-E-M:~/Desktop/C$ gcc menudriven.c
kcr@kcr-To-be-filled-by-0-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):
```

```
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):
```

## (ii) TRANSPOSE

### (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\$