ASSIGNMENT 01 SOLUTIONS (ALL IN ONE)

PROBLEM 1:

In [ ]:

```
// problem 1
// program is written only for integer square matrices
#include<stdio.h>
#include<math.h>

// function to enter the elements of matrice
void input_matrix(int n,int matrix[10][10])
{
    for(int i=1;i<=n;i++)
        {
                for(int j=1;j<=n;j++)
                {
                        scanf("%d",&matrix[i][j]);
                }
        }
}
// function to print the matrice
void print_matrix(int n,int matrix[10][10])
{
        for(int i=1;i<=n;i++)
        {
                for(int j=1;j<=n;j++)
                {
                        printf("%d\t",matrix[i][j]);
                }
                printf("\n");
        }
}
// function for matrices addition
void addition_matrix(int n,int A[10][10],int B[10][10],int ADD[10][10])
{
        for(int i=1;i<=n;i++)
        {
                for(int j=1;j<=n;j++)
                {
                        ADD[i][j]=A[i][j]+B[i][j];
                }
        }
}
// function for matrices subtraction
void subtraction_matrix(int n,int A[10][10],int B[10][10],int SUB[10][10])
{
        for(int i=1;i<=n;i++)
        {
                for(int j=1;j<=n;j++)
                {
                        SUB[i][j]=A[i][j]-B[i][j];
                }
        }
}
// function for matrices multiplication
void product_matrix(int n,int A[10][10],int B[10][10],int PROD[10][10])
{
        for(int i=1;i<=n;i++)
        {
                for(int j=1;j<=n;j++)
                {
                        PROD[i][j]=0;
                        for(int k=1;k<=n;k++)
                        {
                                PROD[i][j]=PROD[i][j]+A[i][k]*B[k][j];
                        }
                }
        }
}
// function for matrice trace
int trace(int n,int matrix[10][10])
{
        int trace=0;
        for(int i=1;i<=n;i++)
        {
                for(int j=1;j<=n;j++)
                {
                        if(i==j)
                        {
                                trace=trace+matrix[i][j];
                        }
                }
        }
        return trace;
}
// function to transfer elements of one matrice to another
void transfer_matrix(int n,int A[10][10],int B[10][10])
{
        for(int i=1;i<=n;i++)
        {
                for(int j=1;j<=n;j++)
                {
                        B[i][j]=A[i][j];
                }
        }
}
// main function to do our job
int main()
{
    int n,A[10][10],B[10][10],C[10][10];

    printf("Enter the dimensions of square matrix:");
    scanf("%d",&n);

    printf("\nEnter the elements of matrix A:\n");
    input_matrix(n,A);
    printf("Matrix A:\n");
    print_matrix(n,A);

    printf("\nEnter the elements of matrix B:\n");
    input_matrix(n,B);
    printf("Matrix B:\n");
    print_matrix(n,B);

        addition_matrix(n,A,B,C);
    printf("\nAddition of matrix A and B:\n");
    print_matrix(n,C);

        subtraction_matrix(n,A,B,C);
    printf("\nSubtraction of matrix A and B:\n");
    print_matrix(n,C);

        product_matrix(n,A,B,C);
    printf("\nMultiplication of matrix A and B:\n");
    print_matrix(n,C);

        printf("\ntrace of matrix A: %d\n",trace(n,A));
        printf("trace of matrix B: %d\n",trace(n,B));

        transfer_matrix(n,A,B);
    printf("\nNew matrix B (elements of A):\n");
    print_matrix(n,B);
}
```

OUTPUT:

```
Enter the dimensions of square matrix:3

Enter the elements of matrix A:
1 2 3 4 5 6 7 8 9
Matrix A:
1  2  3
4  5  6
7  8  9

Enter the elements of matrix B:
9 8 7 6 5 4 3 2 1
Matrix B:
9  8  7
6  5  4
3  2  1

Addition of matrix A and B:
10  10  10
10  10  10
10  10  10

Subtraction of matrix A and B:
-8  -6  -4
-2  0  2
4  6  8

Multiplication of matrix A and B:
30  24  18
84  69  54
138  114  90

trace of matrix A: 15
trace of matrix B: 15
```

New matrix B (elements of A):

1 2 3

4 5 6

7 8 9

Now we need to assemble the above matrice operation functions in a new file " `matops.c` " so that we can use it for the next problem and also for any similiar matrice operations. Creating the file below.

In [ ]:

```
// it contains functions for the matrix operations
// (add,sub,product,trace,transferring elements)
// use #include"matops.c" in the program you wished to use this
#include<stdio.h>
#include<math.h>

// function to print the matrice
void print_matrix(int n,double matrix[10][10])
{
        for(int i=1;i<=n;i++)
        {
                for(int j=1;j<=n;j++)
                {
                        printf("%.2lf\t",matrix[i][j]);
                }
                printf("\n");
        }
}
// function for matrices addition
void addition_matrix(int n,double A[10][10],double B[10][10],double ADD[10][10])
{
        for(int i=0;i<=n;i++)
        {
                for(int j=0;j<=n;j++)
                {
                        ADD[i][j]=A[i][j]+B[i][j];
                }
        }
}
// function for matrices subtraction
void subtraction_matrix(int n,double A[10][10], double B[10][10], double SUB[10]
{
    for (int i=1;i<=n;i++)
    {
        for (int j=1;j<=n;j++)
        {
            SUB[i][j]=A[i][j]-B[i][j];
        }
    }
}
// function for matrices multiplication
void product_matrix(int n,double A[10][10], double B[10][10], double PROD[10][1
{
    for(int i=1;i<=n;i++)
    {
        for(int j=1;j<=n;j++)
        {
            PROD[i][j]=0;
            for(int k=1;k<=n;k++)
            {
                PROD[i][j]=PROD[i][j]+A[i][k]*B[k][j];
            }
        }
    }
}
// function for matrice trace
int trace(int n,double matrix[10][10])
{
        int trace=0;
        for(int i=0;i<=n;i++)
        {
                for(int j=0;j<=n;j++)
                {
                        if(i==j)
                        {
                                trace=trace+matrix[i][j];
                        }
                }
        }
        return trace;
}
// function to transfer the elements of a matrix to another
void transfer_matrix(int n,double A[10][10],double B[10][10])
{
        for(int i=0;i<=n;i++)
        {
                for(int j=0;j<=n;j++)
                {
                        B[i][j]=A[i][j];
                }
        }
}
```

Now whenever we need to perform these operations we'll use this file `#include"matops.c"` as a library.

How to use the functions?

- Print any matrice: `print_matrix(n,A)`
- Addition: `addition_matrix(n,A,B,C)`
- Substraction: `subtraction_matrix(n,A,B,C)`
- Product: `product_matrix(n,A,B,C)`
- Trace of matrice: `trace(n,A)`
- Transfer elemnts of A into B: `transfer_matrix(n,A,B)`

Where **n** is the dimension of square matrice while **A** and **B** are the matrices at which we are applying these operations and **C** is the resultant matrice. All the other parameters are obvious.

PROBLEM 2:

In [ ]:

```
// problem 2
// make sure file matops.c is in same directory
#include<stdio.h>
#include<math.h>
#include"matops.c"

// function to generate the matrices
```

```
void input_matrix(int n,double A[10][10],double B[10][10])
{
    for(int i=1;i<=n;i++)
    {
        for(int j=1;j<=n;j++)
        {
            A[i][j]=(double)i/(double)(i+j);
            B[i][j]=(double)j/(double)(i+j);
        }
    }
}
// main program to calculate the commutator
int main()
{
    int n=4;
    double a[10][10],b[10][10],ab[10][10],ba[10][10],com[10][10];
    input_matrix(n,a,b);

    printf("printing the matrix a:\n");
    print_matrix(n,a);
    printf("\nprinting the matrix b:\n");
    print_matrix(n,b);

    product_matrix(n,a,b,ab);   //commutator 1st term
    product_matrix(n,b,a,ba);   //commutator 2nd term
    subtraction_matrix(n,ab,ba,com);
    printf("\ncommutator c=[a,b]=ab-ba:\n");
    print_matrix(n,com);
}
```

OUTPUT:

```
printing the matrix a:
0.50  0.33  0.25  0.20
0.67  0.50  0.40  0.33
0.75  0.60  0.50  0.43
0.80  0.67  0.57  0.50

printing the matrix b:
0.50  0.67  0.75  0.80
0.33  0.50  0.60  0.67
0.25  0.40  0.50  0.57
0.20  0.33  0.43  0.50

commutator c=[a,b]=ab-ba:
-1.43  -0.82  -0.44  -0.18
-0.82  -0.20  0.18  0.44
-0.44  0.18  0.56  0.82
-0.18  0.44  0.82  1.08
```