Lab manual #9:

Name: Muhammad Ahmed Naseer

Sec: ME-15(A)

CMS id:467276

Task#1:

```
#include<iostream>
using namespace std;
int main()
{
int matrix[3][3];
cout<<"input the numbers for matrix:";
for(int i=0;i<3;i++){
for(int j=0;j<3;j++){
cin>>matrix[i][j];
}
}
int sum1=0;
int sum2=0;
cout<<"the input matrix is:"<<endl;
for(int i=0;i<3;i++){
for( int j=0;j<3;j++){
cout<<matrix[i][j];
if(i==j){
sum1+=matrix[i][j];
}
if(j==3-i-1){
sum2+=matrix[i][j];
}
}
cout<<endl;
}
```

```
cout<<"the sum of right diagonal is:"<<sum1<<endl;

cout<<"the sum of left diagonal is:"<<sum2<<endl;

}
```

```
input the numbers for matrix:
1
2
3
4
5
6
7
8
9
the input matrix is:
123
456
789
the sum of right diagonal is:15
the sum of left diagonal is:15

--------------------------------
Process exited after 17.4 seconds with return value 0
Press any key to continue . . .
```

Task#2:

```
#include<iostream>

using namespace std;


void inputarray (int arr[3][3])

{

for(int i=0;i<3;i++){

for(int j=0;j<3;j++){

cin>>arr[i][j];

}

}
```

```
}
void outputarray (int arr[3][3])
{
for(int i=0;i<3;i++){
for(int j=0;j<3;j++){
cout<<arr[i][j];
}
cout<<endl;
}
}
void sum_array (int a1[3][3], int a2[3][3] , int sum_a[3][3])
{
for(int i=0;i<3;i++){
for(int j=0;j<3;j++){
sum_a[i][j]=a1[i][j] + a2[i][j];
}
}
}
int main()
{
int a1array[3][3] , a2array[3][3] , sum[3][3];
cout<<"First array:"<<endl;
inputarray (a1array);
cout<<"second array:"<<endl;
inputarray (a2array);
cout<<"first array:"<<endl;
outputarray (a1array);
cout<<endl;
cout<<"second array:"<<endl;
```

```
outputarray (a2array);

sum_array(a1array, a2array, sum);

cout<<"the sum of matrices is:"<<endl;

outputarray(sum);

return 0;

}
```

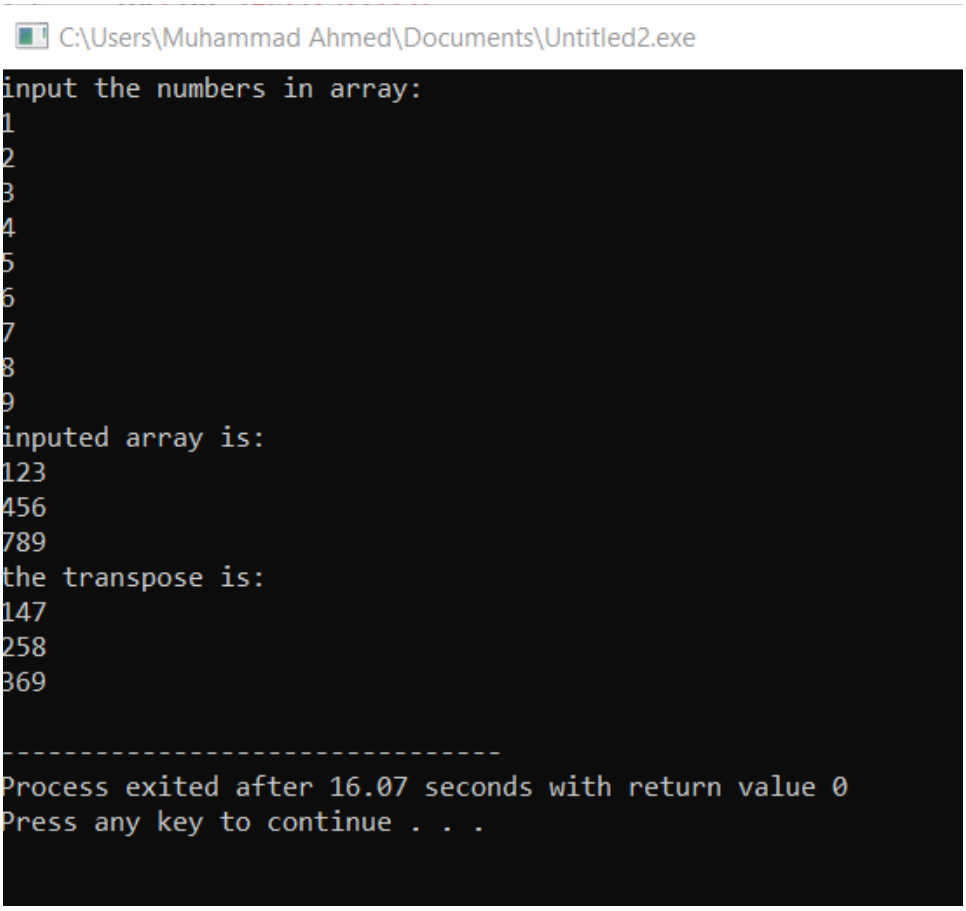
```
First array:
1
2
3
4
5
6
7
8
9
second array:
9
8
7
6
5
4
3
2
1
first array:
123
456
789

second array:
987
654
321
the sum of matrices is:
101010
101010
101010

--------------------------------
Process exited after 58.24 seconds with return value 0
Press any key to continue . . .
```

Task#3:

```
#include<iostream>

using namespace std;


void inputarray (int arr[3][3])
```

```
{
for(int i=0;i<3;i++){
for(int j=0;j<3;j++){
cin>>arr[i][j];
}
}
}
void outputarray (int arr[3][3])
{
for(int i=0;i<3;i++){
for(int j=0;j<3;j++){
cout<<arr[i][j];
}
cout<<endl;
}
}
void transpose(int array[3][3], int transposearray[3][3])
{
for(int i=0; i<3; i++){
for(int j=0;j<3 ;j++){
transposearray[i][j]=array[j][i];
}
}
}
int main()
{
int arr[3][3], t_array[3][3];
cout<<"input the numbers in array:"<<endl;
inputarray(arr);
```

```
cout<<"inputed array is:"<<endl;

outputarray(arr);

transpose(arr , t_array);

cout<<"the transpose is:"<<endl;

outputarray (t_array);

return 0;

}
```



Task#4:

```
#include<iostream>

using namespace std;


void inputarray (int arr[3][3])
```

```
{
for(int i=0;i<3;i++){
for(int j=0;j<3;j++){
cin>>arr[i][j];
}
}
}
void outputarray (int arr[3][3])
{
for(int i=0;i<3;i++){
for(int j=0;j<3;j++){
cout<<arr[i][j]<<" ";
}
cout<<endl;
}
}
void multiply2array ( int a1[3][3], int a2[3][3], int product[3][3])
{
for(int i=0; i<3; i++){
for(int j=0; j<3; j++){
product[i][j]=0;
for(int k=0; k<3; k++){
product[i][j]+=a1[i][k] * a2[k][j];
}
}
}
}
int main()
{
```

```
int arr1[3][3], arr2[3][3], product[3][3];
cout<<"enter numbers in first array:"<<endl;
inputarray(arr1);
cout<<"enter numbers in second array:"<<endl;
inputarray(arr2);
cout<<"array 1:"<<endl;
outputarray(arr1);
cout<<"array 2:"<<endl;
outputarray(arr2);
multiply2array (arr1, arr2, product);
cout<<"the product is"<<endl;
outputarray(product);
return 0;
}
```

```
enter numbers in first array:
1
2
3
4
5
6
7
8
9
enter numbers in second array:
1
2
3
4
5
6
7
8
9
array 1:
1 2 3
4 5 6
7 8 9
array 2:
1 2 3
4 5 6
7 8 9
the product is
30 36 42
66 81 96
102 126 150

--------------------------------
Process exited after 21.6 seconds with return value 0
Press any key to continue . . .
```

Task#5:

```
#include<iostream>

using namespace std;


void multiplicationtable (int num,int limit, int i=1)

{

if(i>limit){return ;}

cout<<num<<"x"<<i<<"="<<num*i<<endl;

multiplicationtable(num, limit, i+1);

}

int main()
```

```
{
int lim;
cout<<"input the number till which table is needed:";
cin>>lim;
cout<<"the table of 15 is:"<<endl;
multiplicationtable(15, lim);
}
```

```
input the number till which table is needed:12
the table of 15 is:
15x1=15
15x2=30
15x3=45
15x4=60
15x5=75
15x6=90
15x7=105
15x8=120
15x9=135
15x10=150
15x11=165
15x12=180

--------------------------------
Process exited after 12.22 seconds with return value 0
Press any key to continue . . .
```

Home Task#1:

```
#include<iostream>
using namespace std;

void inputmatrix (double matrix[3][3])
{
for(int i=0;i<3;i++){
for(int j=0;j<3;j++){
cin>>matrix[i][j];
```

```
}
}
}
void outputmatrix (double matrix[3][3])
{
for(int i=0;i<3;i++){
for(int j=0;j<3;j++){
cout<<matrix[i][j]<<" ";
}
cout<<endl;
}
}
double det2by2 (double a, double b, double c, double d)
{
return a*d-b*c;
}
float det3by3(double matrix[3][3])
{
return matrix[0][0]*det2by2( matrix[1][1], matrix[1][2], matrix[2][1], matrix[2][2])-
        matrix[0][1]*det2by2( matrix[1][0], matrix[1][2], matrix[2][0], matrix[2][2])+
        matrix[0][0]*det2by2( matrix[1][0], matrix[1][1], matrix[2][0], matrix[2][1]);
}
void adjoint (double matrix[3][3], double adjmatrix[3][3])
{
adjmatrix[0][0]=+det2by2(matrix[1][1], matrix[1][2], matrix[2][1], matrix[2][2]);
adjmatrix[0][1]=-det2by2(matrix[0][1], matrix[0][2], matrix[2][1], matrix[2][2]);
adjmatrix[0][2]=+det2by2(matrix[0][1], matrix[0][2], matrix[1][1], matrix[1][2]);
adjmatrix[1][0]=+det2by2(matrix[1][0], matrix[1][2], matrix[2][0], matrix[2][2]);
adjmatrix[1][1]=+det2by2(matrix[0][0], matrix[0][2], matrix[2][0], matrix[2][2]);
```

```
adjmatrix[1][2]=+det2by2(matrix[0][0], matrix[0][2], matrix[1][0], matrix[1][2]);
adjmatrix[2][0]=+det2by2(matrix[1][0], matrix[1][1], matrix[2][0], matrix[2][1]);
adjmatrix[2][1]=+det2by2(matrix[0][0], matrix[0][1], matrix[2][0], matrix[2][1]);
adjmatrix[2][2]=+det2by2(matrix[0][0], matrix[0][1], matrix[1][0], matrix[1][1]);
}
double inverse (double matrix[3][3], double inverse[3][3])
{
int det=det3by3(matrix);
   if(det==0){
cout<<"matrix is singular,no inverse."<<endl;
}
else{
double adj[3][3];
adjoint (matrix, adj);
for(int i=0;i<3;i++){
for(int j=0;j<3;j++){
inverse[i][j]=adj[i][j]/det;
}
 }
}
}
int main()
{
double matrix[3][3];
cout<<"input numbers of matrix:"<<endl;
inputmatrix(matrix);
cout<<"the matrix is:"<<endl;
outputmatrix(matrix);
double inv[3][3];
```

```
inverse(matrix,inv);
cout<<"the inverse of matrix is:"<<endl;
outputmatrix(inv);
return 0;
}
```

```
input numbers of matrix:
1
2
3
4
5
6
7
8
9
the matrix is:
1 2 3
4 5 6
7 8 9
the inverse of matrix is:
-0.5 1 -0.5
-1 -2 -1
-0.5 -1 -0.5

--------------------------------
Process exited after 20.13 seconds with return value 0
Press any key to continue . . .
```