



CS-114 - Fundamental of Programing

Lab AND Home Tasks 9

Course Instructor: Dr Jawad Khan

Lab Instructor: Muhammad Affan

Student Name: Ahmed Adil Hussain

CMS ID: 477537

DATE:

6 December 2023

**LAB TASK 1:**

1. Make 2D Array in C++ and print left diagonal and right diagonal sum of a 3x3 matrix.

Code:

```
1  #include <iostream>
2
3  using namespace std;
4  int main()
5  {
6      // LAB TASK 1
7      int mat [3][3];
8      cout<<"Please enter the elements for the matrix: "<<endl;
9      for(int j=0; j<3; j++){
10         for(int i=0; i<3; i++){cin>>mat[i][j];}
11     }
12     int lsum, rsum;
13     lsum = mat[0][0] + mat[1][1] + mat[2][2];
14     rsum = mat[2][0] + mat[1][1] + mat[0][2];
15
16     for(int j=0; j<3; j++){
17         for(int i=0; i<3; i++){
18             cout<<mat[i][j]<<" ";
19         }
20     }
21     cout<<endl;
22     cout<<"The left diagonal sum is "<<lsum<<", and the right diagonal sum is "<<rsum<<".";
23
24     return 0;
25 }
```

Result:

```
Please enter the elements for the matrix:
1
2
3
4
5
6
7
8
9
1 2 3
4 5 6
7 8 9
The left diagonal sum is 15, and the right diagonal sum is 15.
-----
Process exited after 3.134 seconds with return value 0
Press any key to continue . . .
```

**LAB TASK 2:**

2. Write a function to add two 2D arrays of size 3x3.

Code:

```

1  #include <iostream>
2
3  using namespace std;
4
5  void matrixsum(int a[3][3], int b[3][3]){
6      int sum[3][3];
7      for(int j=0; j<3; j++){
8          for(int i=0; i<3; i++){
9              sum[i][j] = a[i][j] + b[i][j];
10             }
11         }
12         for(int j=0; j<3; j++){
13             for(int i=0; i<3; i++){
14                 cout<<a[i][j]<<" ";
15             }
16             cout<<endl;
17         }
18         cout<<endl<<"+"<<endl;
19         for(int j=0; j<3; j++){
20             for(int i=0; i<3; i++){
21                 cout<<b[i][j]<<" ";
22             }
23             cout<<endl;
24         }
25         cout<<endl<<"="<<endl;
26         for(int j=0; j<3; j++){
27             for(int i=0; i<3; i++){
28                 cout<<sum[i][j]<<" ";
29             }
30             cout<<endl;
31         }
32     }
33
34     int main()
35     {
36         // LAB TASK 2
37         int a [3][3];
38         int b [3][3];
39         int sum [3][3];
40         cout<<"Please enter the elements for the first matrix: "<<endl;
41         for(int j=0; j<3; j++){
42             for(int i=0; i<3; i++){cin>>a[i][j];}
43         }
44         cout<<"Please enter the elements for the second matrix: "<<endl;
45         for(int j=0; j<3; j++){
46             for(int i=0; i<3; i++){cin>>b[i][j];}
47         }
48
49         cout<<"sum matrix using sum function: "<<endl;
50         matrixsum(a, b);
51
52         return 0;
53     }
54 }

```

Result:

```

Please enter the elements for the first matrix:
1
2
3
4
5
6
7
8
9
Please enter the elements for the second matrix:
1
2
3
4
5
6
7
8
9
sum matrix using sum function:
1 2 3
4 5 6
7 8 9

+
1 2 3
4 5 6
7 8 9

=
2 4 6
8 10 12
14 16 18
-----

```

**LAB TASK 3:**

3. Using 2D arrays in C++, take transpose of a 3x3 matrix. Make a transpose function

Code:

```
1  #include <iostream>
2  using namespace std;
3
4  void transpose(int a[3][3]){
5      cout<<"Given matrix: "<<endl;
6      for(int j=0; j<3; j++){
7          for(int i=0; i<3; i++){
8              cout<<a[i][j]<<" ";
9          }
10         cout<<endl;
11     }
12     cout<<"Transposed matrix: "<<endl;
13     for(int i=0; i<3; i++){
14         for(int j=0; j<3; j++){
15             cout<<a[i][j]<<" ";
16         }
17         cout<<endl;
18     }
19
20 int main()
21 {
22     // LAB TASK 3
23     int a[3][3];
24     cout<<"Please enter the elements for the matrix: "<<endl;
25     for(int j=0; j<3; j++){
26         for(int i=0; i<3; i++){cin>>a[i][j];}
27     }
28     cout<<"Transposed matrix using transpose function: "<<endl;
29     transpose(a);
30
31
32     return 0;
33 }
```

Result:

```
Please enter the elements for the matrix:
1
2
3
4
5
6
7
8
9
Transposed matrix using transpose function:
Given matrix:
1 2 3
4 5 6
7 8 9
Transposed matrix:
1 4 7
2 5 8
3 6 9

-----
Process exited after 3.244 seconds with return value 0
Press any key to continue . . .
```

**LAB TASK 4:**

4. Using 2D arrays in C++, implement 3x3 matrix multiplication. Make a function.

Code:

```

1  #include <iostream>
2  using namespace std;
3
4  void matrixproduct(int a[3][3], int b[3][3]){
5      int product[3][3];
6      for(int i=0; i<3; i++){
7          for(int j=0; j<3; j++){
8              product[i][j]=0;
9              for(int p=0; p<3; p++){
10                 product[i][j]=product[i][j] + a[p][j]*b[i][p];
11             }
12         }
13     }
14     cout<<"Product matrix: "<<endl;
15     for(int j=0; j<3; j++){
16         for(int i=0; i<3; i++){
17             cout<<product[i][j]<<" ";
18         }
19         cout<<endl;
20     }
21 }
22
23 int main()
24 {
25     // LAB TASK 4
26     int a [3][3];
27     int b [3][3];
28     int product[3][3];
29     cout<<"Please enter the elements for the first matrix: "<<endl;
30     for(int j=0; j<3; j++){
31         for(int i=0; i<3; i++){cin>>a[i][j];}
32     }
33     cout<<"Please enter the elements for the second matrix: "<<endl;
34     for(int j=0; j<3; j++){
35         for(int i=0; i<3; i++){cin>>b[i][j];}
36     }
37     cout<<"a: "<<endl;
38     for(int j=0; j<3; j++){
39         for(int i=0; i<3; i++){
40             cout<<a[i][j]<<" ";
41         }
42         cout<<endl;
43     }
44     cout<<"b: "<<endl;
45     for(int j=0; j<3; j++){
46         for(int i=0; i<3; i++){
47             cout<<b[i][j]<<" ";
48         }
49         cout<<endl;
50     }
51     cout<<"Product matrix using product function: "<<endl;
52     matrixproduct(a, b);
53     return 0;
54 }

```

Result:

```

Please enter the elements for the first matrix:
1
2
3
4
5
6
7
8
9
Please enter the elements for the second matrix:
1
2
3
4
5
6
7
8
9
a:
1 2 3
4 5 6
7 8 9
b:
1 2 3
4 5 6
7 8 9
Product matrix using product function:
Product matrix:
30 36 42
66 81 96
102 126 150
-----
Process exited after 5.149 seconds with return value 0

```

**LAB TASK 5:**

5. Print the multiplication table of 15 using recursion.

Code:

```
1  #include <iostream>
2  using namespace std;
3
4  void table(int num, int m, int d=1)
5  {
6      if (d>m){return ;}
7      cout<<num<<" x "<<d<<" = "<<num*d<<endl;
8      table(num, m, d+1);
9  }
10
11
12  int main()
13  {
14      // LAB TASK 5
15      int m;
16      cout<<"Please enter the max number of multiples: ";
17      cin>>m;
18      cout<<"The multiplication table of 15 is: "<<endl;
19      table(15, m);
20
21
22      return 0;
23  }
```

Result:

```
Please enter the max number of multiples: 12
The multiplication table of 15 is:
15 x 1 = 15
15 x 2 = 30
15 x 3 = 45
15 x 4 = 60
15 x 5 = 75
15 x 6 = 90
15 x 7 = 105
15 x 8 = 120
15 x 9 = 135
15 x 10 = 150
15 x 11 = 165
15 x 12 = 180
```

```
-----
Process exited after 1.897 seconds with return value 0
Press any key to continue . . .
```

**HOME TASK:**

Write a C++ program to take inverse of a 3x3 matrix using its determinant and adjoint..

Code: (result next page)

```

1  #include <iostream>
2
3  using namespace std;
4  int main()
5  {
6      int mat[3][3];
7      float adj[3][3];
8      float inv[3][3];
9      int det;
10     cout<<"Please enter the elements for the matrix: "<<endl;
11     for(int j=0; j<3; j++){
12         for(int i=0; i<3; i++){cin>>mat[i][j];}
13     }
14     cout<<"Given matrix: "<<endl;
15     for(int j=0; j<3; j++){
16         for(int i=0; i<3; i++){
17             cout<<mat[i][j]<<" ";
18         }
19         cout<<endl;
20     }
21     int a=mat[0][0],b=mat[0][1],c=mat[0][2],d=mat[1][0],e=mat[1][1],f=mat[1][2],g=mat[2][0],h=mat[2][1],i=mat[2][2];
22     // determinant calculation
23     // det = 00(11.22-21.12) - 10(01.22-21.02) + 20(01.12-11.02)
24     det = a*(e*i - f*h) - b*(d*i - f*g) + c*(d*h - e*g);
25     if(det==0){cout<<"This matrix has no inverse."; return 0;}
26
27     // inv = 1/d (adjoint)
28     // adjoint calculation
29     adj[0][0] = e*i - f*h;
30     adj[0][1] = -1*(b*i - c*h);
31     adj[0][2] = b*f - c*e;
32     adj[1][0] = -1*(d*i - f*g);
33     adj[1][1] = a*i - c*g;
34     adj[1][2] = -1*(a*f - c*d);
35     adj[2][0] = d*h - e*g;
36     adj[2][1] = -1*(a*h - b*g);
37     adj[2][2] = a*e - b*d;
38
39     for(int j=0; j<3; j++){
40         for(int i=0; i<3; i++){
41             inv[i][j]=adj[i][j]/det;
42         }
43     }
44     cout<<"Inverse matrix: "<<endl;
45     for(int j=0; j<3; j++){
46         for(int i=0; i<3; i++){
47             cout<<inv[i][j]<<" ";
48         }
49         cout<<endl;
50     }
51     return 0;
52 }

```



Result:

```
Please enter the elements for the matrix:
5
3
1
1
2
1
3
4
1
Given matrix:
5 3 1
1 2 1
3 4 1

inverse matrix:
0.333333 -0.166667 -0.166667
-0.333333 -0.333333 0.666667
0.333333 1.833333 -1.16667

-----
Process exited after 5.308 seconds with return value 0
Press any key to continue . . .
```