

# Fundamentals of programming

## Lab Manual 9



*Muhammad Abdullah*

*ME-15 Section A*

*Qalam: 454502*

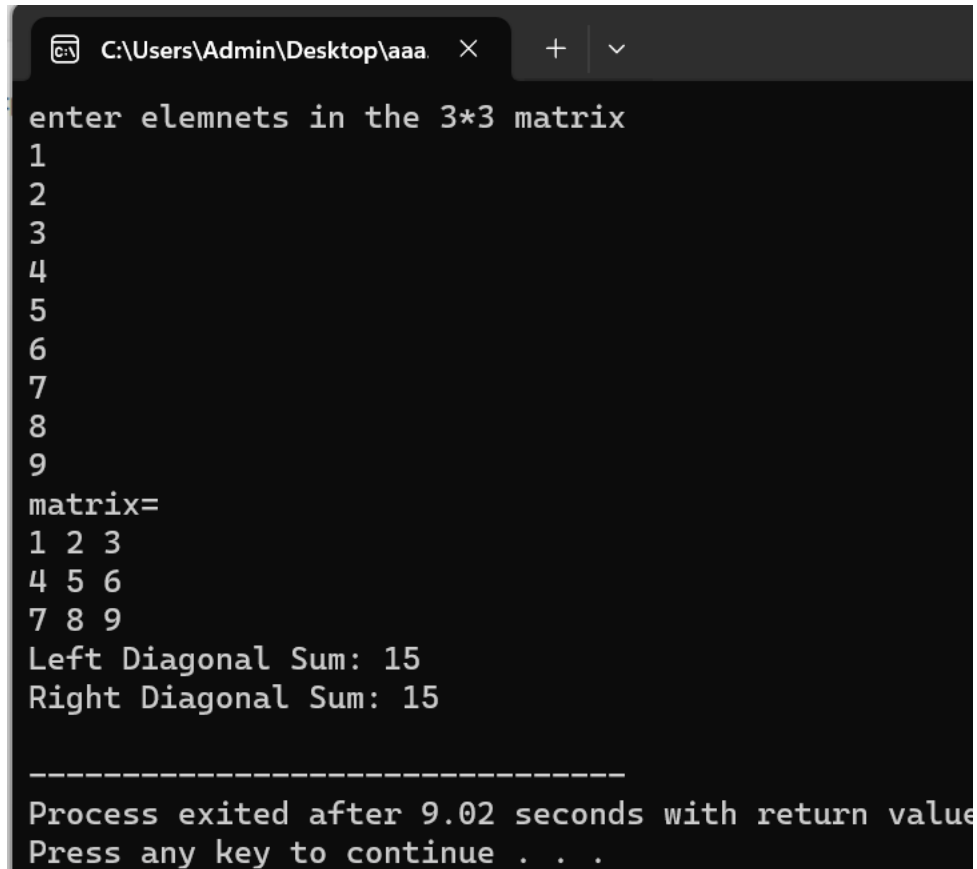
## Lab Task:

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

```
int main(){
    int i, j, num;
    int matrix[3][3];
    int lDSum, rDSum = 0;
    cout<<"enter elements in the 3*3 matrix"<<endl;
    for (i=0; i<3; i++){
        for(j=0; j<3; j++){
            cin>>matrix[i][j];
        }
    }
    cout<<"matrix= "<<endl;
    for (i=0; i<3; i++){
        for(j=0; j<3; j++){
            cout<<matrix[i][j]<<" ";
        }
        cout<<endl;
    }
    for ( i = 0; i < 3; ++i) {
        lDSum += matrix[i][i];
    }
    cout << "Left Diagonal Sum: " << lDSum <<endl;
    for (i=2; i>=0; i--){
        rDSum+=matrix[i][i];
    }
}
```

```
cout << "Right Diagonal Sum: " << rDSum << endl;
```

```
}
```



```
C:\Users\Admin\Desktop\aaa
enter elemnets in the 3*3 matrix
1
2
3
4
5
6
7
8
9
matrix=
1 2 3
4 5 6
7 8 9
Left Diagonal Sum: 15
Right Diagonal Sum: 15
-----
Process exited after 9.02 seconds with return value
Press any key to continue . . .
```

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

```
int main(){
int arr1[3][3];
int arr2[3][3];
int sum[3][3];

cout<<"enter elemnets of array 1"<<endl;
for (int i=0;i<3;i++){
    for(int j=0;j<3;j++){
        cin>>arr1[i][j];
    }
}
```

```

    }

    cout<<"enter elemnets of array 2"<<endl;
    for (int i=0;i<3;i++){
        for(int j=0;j<3;j++){
            cin>>arr2[i][j];
        }
    }

    cout<<"array 1 = "<<endl;
    for (int i=0;i<3;i++){
        for(int j=0;j<3;j++){
            cout<<arr1[i][j]<<" ";
        }
        cout<<endl;
    }

    cout<<"array 2 = "<<endl;
    for (int i=0;i<3;i++){
        for(int j=0;j<3;j++){
            cout<<arr2[i][j]<<" ";
        }
        cout<<endl;
    }

    cout<<"sum = "<<endl;
    for(int x=0;x<3;x++){
        for(int y=0;y<3;y++){
            sum[x][y]=arr1[x][y]+arr2[x][y];
            cout<<sum[x][y]<<" ";
        }
    }

```

```
    cout<<endl;
}
```

```
D:\C++\lab Manual 9.exe
6
7
8
9
enter elemnets of array 2
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
sum =
2 4 6
8 10 12
14 16 18

-----
Process exited after 14.52 seconds with return
Press any key to continue . . . |
n results...
```

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

```
void transpose(int num[3][3]){
    int temp;
```

```

int transp[3][3];
for(int i=0;i<3;i++){
    for(int j=0;j<3;j++){
        transp[i][j]=num[j][i];
    }
}
cout<<"transpose= "<<endl;
for(int x=0;x<3;x++){
    for(int y=0;y<3;y++){
        cout<<transp[x][y]<<" ";
    }
    cout<<endl;
}
}

int main(){
    int arr[3][3];
    cout<<"enter elements of array you want to find transpose of "<<endl;
    for(int x=0;x<3;x++){
        for(int y=0;y<3;y++){
            cin>>arr[x][y];
        }
    }
    cout<<"matrix= "<<endl;
    for(int x=0;x<3;x++){
        for(int y=0;y<3;y++){
            cout<<arr[x][y]<<" ";
        }
    }
}

```

```

        cout<<endl;
    }
    transpose(arr);
}

```

```

C:\Users\Admin\Desktop\aaa. X + v
enter elements of array you want to find transpose of
1
2
3
4
5
6
7
8
9
matrix=
1 2 3
4 5 6
7 8 9
transpose=
1 4 7
2 5 8
3 6 9

-----
Process exited after 8.232 seconds with return value 0
Press any key to continue

```

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

```

void multiply(int num1[3][3],int num2[3][3]){
    int result[3][3];
    for (int i = 0; i < 3; i++) {
    for (int j = 0; j < 3; j++) {
        result[i][j] = 0;

```

```

        for (int k = 0; k < 3; k++) {
            result[i][j] += num1[i][k] * num2[k][j];
        }
    }
}

        for(int x=0;x<3;x++){
            for(int y=0;y<3;y++){
                cout<<result[x][y]<<" ";
            }
            cout<<endl;
        }
}

int main(){
    int arr1[3][3],arr2[3][3];
    cout<<"enter elemnets of array 1"<<endl;
    for (int i=0;i<3;i++){
        for(int j=0;j<3;j++){
            cin>>arr1[i][j];
        }
    }

    cout<<"enter elemnets of array 2"<<endl;
    for (int i=0;i<3;i++){
        for(int j=0;j<3;j++){
            cin>>arr2[i][j];
        }
    }

    cout<<"array 1 = "<<endl;

```



```
        for (int i=0;i<3;i++){
            for(int j=0;j<3;j++){
                cout<<arr1[i][j]<<" ";
            }
            cout<<endl;
        }
        cout<<"array 2 = "<<endl;
        for (int i=0;i<3;i++){
            for(int j=0;j<3;j++){
                cout<<arr2[i][j]<<" ";
            }
            cout<<endl;
        }
        cout<<"product= "<<endl;
        multiply(arr1,arr2);
    }
```

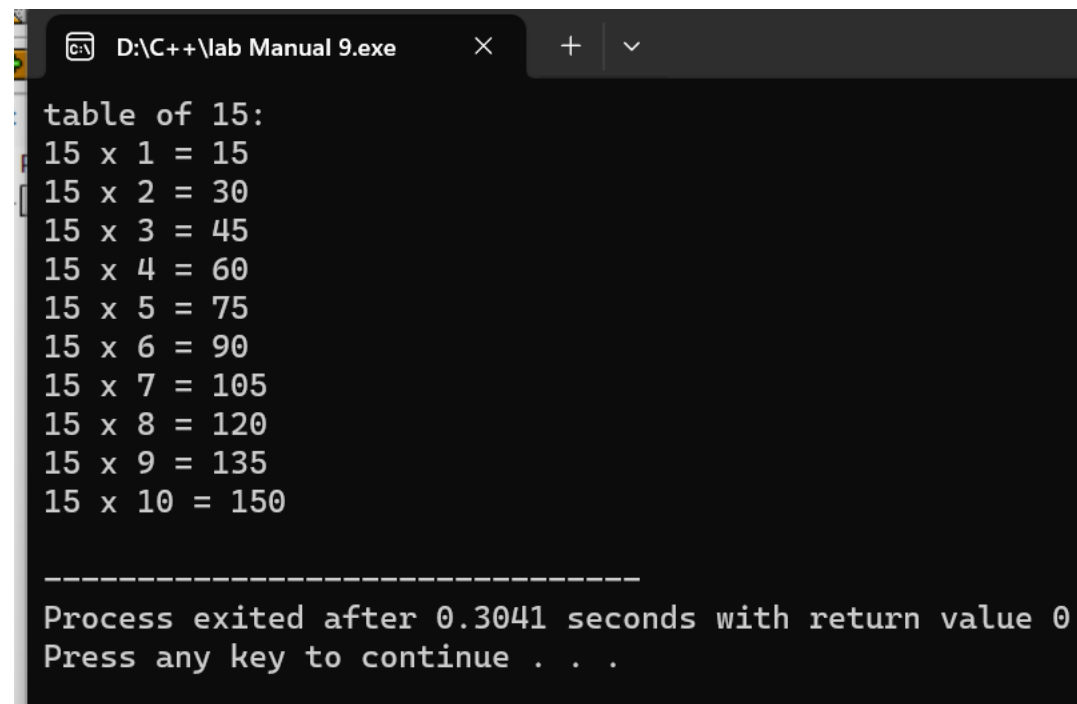
```
C:\Users\Admin\Desktop\aaa  X + v
6
7
8
9
enter elemnets of array 2
1
2
3
4
5
6
7
8
9
array 1 =
1 2 3
4 5 6
7 8 9
array 2 =
1 2 3
4 5 6
7 8 9
product=
30 36 42
66 81 96
102 126 150
-----
Process exited after 15.28 seconds with return
```

**5. Print the multiplication table of 15 using recursion.**

```
void table(int multi,int lim){
    int product;
    if(multi>lim){
        return;
    }

    product=15*multi;
```

```
        cout<<"15 x "<<multi<<" = "<<product<<endl;
        table(multi+1,lim);
    }
int main(){
    int limit=10;
    cout<<"table of 15:"<<endl;
    table(1,limit);
}
```

A screenshot of a Windows command prompt window titled "D:\C++\lab Manual 9.exe". The window displays the output of a C++ program. It first prints "table of 15:" followed by a list of multiplication results from 15 x 1 to 15 x 10. Below this, a horizontal line is printed, followed by the message "Process exited after 0.3041 seconds with return value 0" and "Press any key to continue . . .".

```
table of 15:
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

-----
Process exited after 0.3041 seconds with return value 0
Press any key to continue . . .
```