**Heaven’s Light is Our Guide**



**Rajshahi University of Engineering and Technology**

**Department of Computer Science and Engineering**

**Course No:** CSE.2104

**Course Title:** Sessional based on CSE.2104 ( Numerical Methods )

**Lab Report On:** Triangular Matrices

**Submitted To**

Sadia Zaman Mishu, Assistant Professor, Dept . of CSE RUET

Md. Farukuzzaman Faruk, Lecturer, Dept. of CSE, RUET

**Submitted By**

Md. Ariful Islam  
Roll No: 1803046  
Section: A  
Department: CSE

Date : 21/11/2020

Chapter

1

**Title 1:** Implementation of Numerical Linear Algebra to Find whether a Non-Singular Square Matrix is Triangular or Not.

## **Objective**

* Gathering knowledge about Triangular Matrix.
* Implementing the Knowledge in C++.

## **Methodology**

* Take input of n, of a n\*n square matrix.
* Load the matrix from .txt file.
* Find out whether the matrix is lower or upper or non-triangular matrix -
  + - * + Lower If the elements above the main diagonal are zero.
        + Upper If the elements below the main diagonal are zero.
        + Otherwise the matrix is a non-triangular matrix.

## **Implementation**

I have implemented Numerical Linear Algebra to find the matrix is lower or upper or non-triangular, according to the above Pseudocode. I have taken the matrix from a text file. The tools I used here are :

* + - * + C++
        + Text File
        + Editor: CodeBlocks

### **Code**

|  |
| --- |
| // This code checks whether a given  // non-singular square matrix is  // Lower Triangular  // Upper Triangular  // or Non-Triangular.  #include<bits/stdc++.h>  using namespace std;  double x[51][51];  int n;  void input(){  cout<<"\n\tEnter N\*N Matrix's N: ";  cin>>n;  int v;  freopen("Matrix.txt","r",stdin);  for(int i=0;i<n;i++){  for(int j=0;j<n;j++){  cin>>v;  x[i][j]=v;  }  }  }  void show(){  cout<<"\n\tThe Matrix is :"<<endl;  for(int i=0;i<n;i++){  cout<<"\t";  for(int j=0;j<n;j++){  cout<<" "<<x[i][j];  }  cout<<endl;  }  }  int islow(){  int z=1;  for(int i=0;i<n;i++){  for(int j=i+1;j<n;j++){  if(x[i][j]!=0){  z=0;  break;  }  }  if(!z){  break;  }  }  return z;  }  int isup(){  int z=1;  for(int i=0;i<n;i++){  for(int j=i+1;j<n;j++){  if(x[j][i]!=0){  z=0;  break;  }  }  if(!z){  break;  }  }  return z;  }  void check(){  int low,up;  low=islow();  up=isup();  if(low||up){  if(low&&up){  cout<<"\n\tBoth Lower & Upper Triangular Matrix"<<endl;  }  else if(low){  cout<<"\n\tLower Triangular Matrix"<<endl;  }  else if(up){  cout<<"\n\tUpper Triangular Matrix"<<endl;  }  }  else{  cout<<"\n\tNon-Triangular Matrix"<<endl;  }  }  int main(){  input();  show();  check();  return 0;  } |

## **Output**

I had used the following matrix in the implementation:

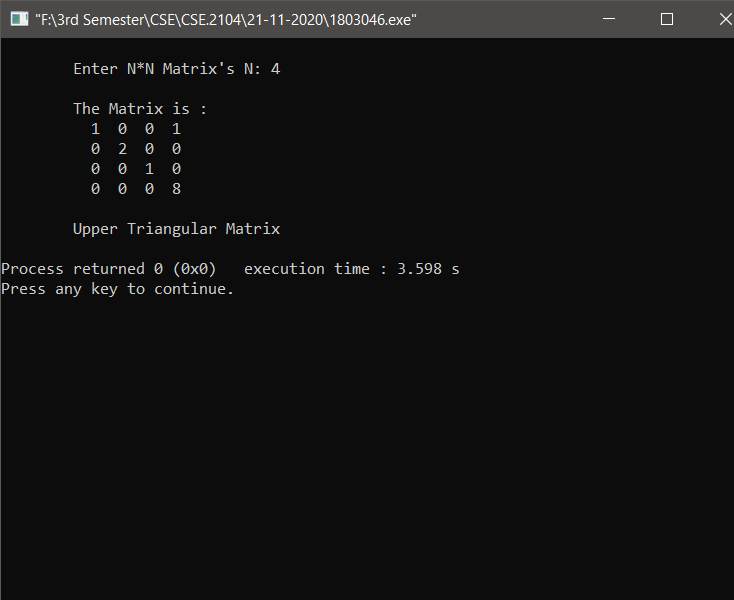
**1 0 0 1**

**0 2 0 0**

**0 0 1 0**

**0 0 0 8**

And my output was like below:



**# End #**