Special Matrix

```
#include <iostream>

#include <fstream>

#include <cmath>

using namespace std;


//a class for the performing various operations on matrix

class matrix

{

private:

   int **m;

   int *a;

   int n = 0; //stores the value of n of nxn matrix

   int s = 0; //stores the size of the array a

   int *arr;


public:

   matrix(){};

   ~matrix(){};


   void input();              //takes input for matrix from the file

   void checkIfSpecial();      //checks if the matrix is special or not

   void matrixToArray(char ch); //converts matrix to 1D array

   void arrayToMatrix();       //converts array to the special matrix

};


//method to take input for matrix from the input file

void matrix::input()

{
```

```
/**************************FILE HANDLING************************************/
//creating an object of ifstream class
ifstream inFile;

//opening the input file to take set data from it
inFile.open("Specialmatrix.txt");

//checking if the input file opens successfully or not
if (!inFile)
{
    cerr << "Error opening in file 1\n";
    exit(100);
}
/***************************************************************************/

//taking input for matrix size from the text file
inFile >> n;

//allocating memory to the 2d array of matrix dynamically
m = new int *[n];
for (int i = 0; i < n; i++)
{
    m[i] = new int[n];
}

//taking input for the matrix from the file
for (int i = 0; i < n; i++)
{
    for (int j = 0; j < n; j++)
    {
        inFile >> m[i][j];
        }
    }

    //printing the Matrix of the graph
    cout << ">>>Given Matrix is: " << endl;
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n; j++)
        {
            cout << m[i][j] << "\t";
        }
        cout << endl;
    }

    /****************************CLOSING FILES*************************************/
    inFile.close();
    //checking file closes or not
    if (inFile.fail())
    {
        cerr << "Error in closing file 2\n";
        exit(102);
    }
    /********************************************************************************/
}

//method to check if the given matrix is special or not
void matrix::checkIfSpecial()
{
    int t = 0;  //to count the elements for tridiagonal matrix
    int lt = 0; //to count the elements for lower triangular matrix
    int ut = 0; //to count the elements for upper triangular matrix
```

```
for (int i = 0; i < n; i++)
{
    for (int j = 0; j < n; j++)
    {
        //checks if the matrix is any Special Matrix
        if (i > j)
        {
            if (i - j == 1)
            {
                if (m[i][j] != 0)
                {
                    ut--;
                    lt++;
                    t++;
                }
            }
            else
            {
                if (m[i][j] != 0)
                {
                    ut--;
                    lt++;
                    t--;
                }
            }

            //check if the matrix is symmetric or not
            if (m[i][j] == m[j][i])
            {
                s++;
    }
}
else if (i < j)
{
    if (j - i == 1)
    {
        if (m[i][j] != 0)
        {
            ut++;
            lt--;
            t++;
        }
    }
    else
    {
        if (m[i][j] != 0)
        {
            ut++;
            lt--;
            t--;
        }
    }
}
else
{
    if (m[i][j] != 0)
    {
        lt++;
        ut++;
        t++;
    }
```

```
            }
        }
    }

    if (s == ((n * n - n) / 2))
    {
        cout << "\n<||> The given matrix is a Symmetric matrix" << endl;
        //we can vonvert symmetric matrix to 1D array as same as upper/lower traingular matrix
        matrixToArray('u');
    }
    else if (t == (3 * n - 2))
    {
        cout << "\n<||> The given matrix is a Tridiagonal Matrix" << endl;
        matrixToArray('t');
    }
    else if (ut == (n * (n + 1) / 2))
    {
        cout << "\n<||> The given Matrix is a Upper Triangular Matrix" << endl;
        matrixToArray('u');
    }
    else if (lt == (n * (n + 1) / 2))
    {
        cout << "\n<||> The given matrix is a Lower Triangular Matrix" << endl;
        matrixToArray('l');
    }
    else
    {
        cout << "\n<||> The given matrix is not a Special Matrix" << endl;
    }
}
```

```
//method to convert matrix into 1D array
void matrix::matrixToArray(char ch)
{
  cout << "\n***| Converting into 1D Array |***\n";
  if (ch == 't')
  {
    s = 3 * n - 2;
    a = new int[s];

    for (int i = 0; i < n; i++)
    {
      for (int j = 0; j < n; j++)
      {
        if (i - j == 1)
        {
          a[i - 1] = m[i][j];
        }

        if (i == j)
        {
          a[n - 1 + i] = m[i][j];
        }

        if (j - i == 1)
        {
          a[2 * n - 1 + i] = m[i][j];
        }
      }
    }
  }
  else if (ch == 'u')
```

```
{
    s = n * (n + 1) / 2;
    a = new int[s];

    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n; j++)
        {
            if (i <= j)
            {
                a[(n * i) - (i * (i + 1) / 2) + j] = m[i][j];
            }
        }
    }
}
else if (ch == 'l')
{
    s = n * (n + 1) / 2;
    a = new int[s];

    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n; j++)
        {
            if (i >= j)
            {
                a[i * (i + 1) / 2 + j] = m[i][j];
            }
        }
    }
}
```

```
  cout << "***Your final 1D Array is :" << endl;
  for (int i = 0; i < s; i++)
  {
    cout << a[i] << "\t";
  }
  cout << endl;
}

//method to change the array into the special matrix
void matrix::arrayToMatrix()
{
  int size = 0;
  char c, selc;

  do
  {
    char x;
    cout << "\n***| Converting Array into special Matrix |***" << endl;
    cout << "\n!!! Do you want to enter another Array or want to use the above? Press (Y/y) if yes 
otherwise type anything : ";
    cin >> x;

    if (x == 'y' || x == 'Y')
    {
      cout << "Enter the Size of your new Array : ";
      cin >> size;

      arr = new int[size];

      cout << "Enter the elements of your Array one by one:\n";
```

```
        for (int i = 0; i < size; i++)
        {
            cin >> arr[i];
        }
        cout << "Your Array has been created successfully" << endl;

        cout << "\n>>>Your New array is: " << endl;
        for (int i = 0; i < size; i++)
        {
            cout << arr[i] << "\t";
        }
        cout << endl;
    }
    else
    {
        arr = a;
        size = s;
    }

    cout << "Enter the type of Special matrix you want to form from the array:\n1. t => tridiagonal\n2. u => upper triangular\n3. l => lower triangular.\n4. s => Symmetric\n";
    cin >> c;

    switch (c)
    {
    case 't':
    {
        //denotes the numver of elements in the row and column of the matrix
        int e = (size + 2) / 3;

        if (size == 3 * e - 2)
```

```
{
  int mat[e][e] = {0};

  //converting the array into Tridiagonal Matrix
  for (int i = 0; i < e; i++)
  {
    for (int j = 0; j < e; j++)
    {
      if (i - j == 1)
      {
        mat[i][j] = arr[i - 1];
      }
      else if (i == j)
      {
        mat[i][j] = arr[n - 1 + i];
      }
      else if (j - i == 1)
      {
        mat[i][j] = arr[(2 * n - 1) + i];
      }
      else
      {
        mat[i][j] = 0;
      }
    }
  }

  cout << "\nTridiagonal Matrix formed from the array is: " << endl;
  for (int x = 0; x < e; x++)
  {
    for (int y = 0; y < e; y++)
```

```
                    {
                        cout << mat[x][y] << "\t";
                    }
                    cout << endl;
                }
            }
            else
            {
                cout << "\n!!! Elements in the Array are greater or smaller than the elements required to form an Tridiagonal Matrix !!!" << endl;
            }
        }
        break;

        case 'u':
        {
            int e = (-1 + sqrt(1 + 8 * size)) / 2;

            if (size == (e * (e + 1) / 2))
            {
                int mat[e][e] = {0};

                //converting the array into Upper triangular Matrix
                for (int i = 0; i < e; i++)
                {
                    for (int j = 0; j < e; j++)
                    {
                        if (i <= j)
                            mat[i][j] = arr[(n * i) - i * (i + 1) / 2 + j];
                        else
                            mat[i][j] = 0;
```

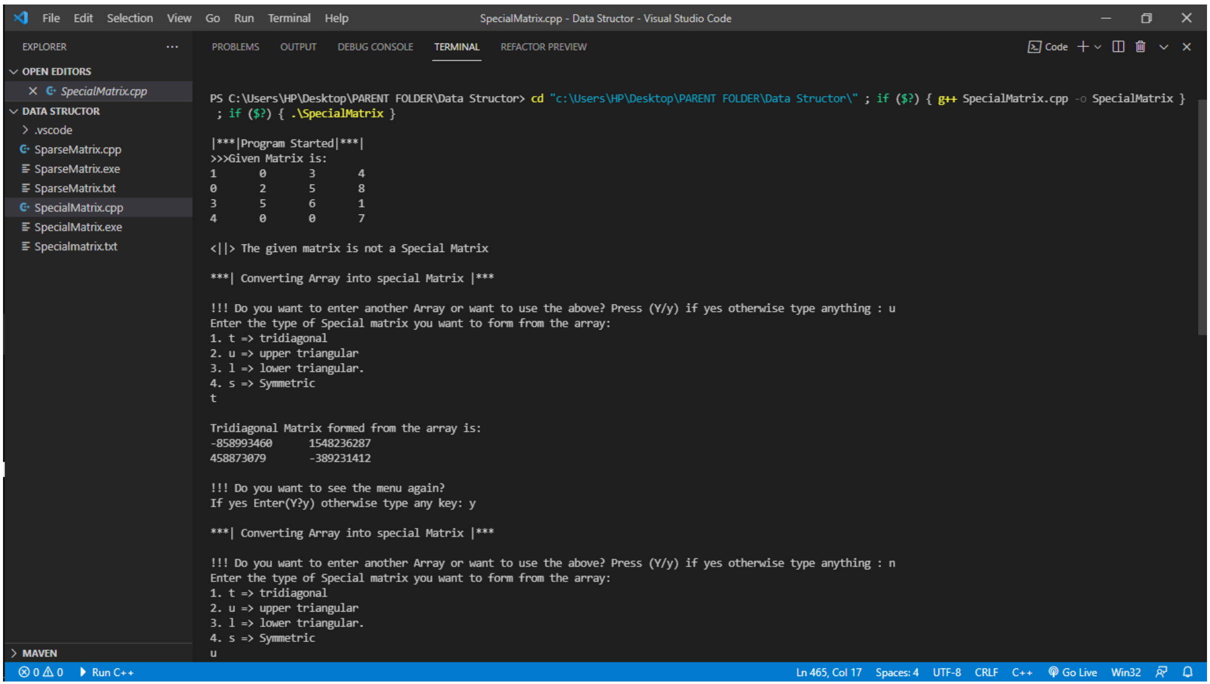
```
          }
        }

        cout << "\nUpper Triangular Matrix formed from the array is: " << endl;
        for (int x = 0; x < e; x++)
        {
          for (int y = 0; y < e; y++)
          {
            cout << mat[x][y] << "\t";
          }
          cout << endl;
        }
      }
      else
      {
        cout << "\n!!! Elements in the Array are more/less than the elements required to form an upper triangular matrix !!!" << endl;
      }
    }
    break;

    case 'l':
    {
      int e = (-1 + sqrt(1 + 8 * size)) / 2;

      if (size == (e * (e + 1) / 2))
      {
        int mat[e][e] = {0};

        //converting the array into lower triangular Matrix
        for (int i = 0; i < e; i++)
```

```
                {
                    for (int j = 0; j < e; j++)
                    {
                        if (i >= j)
                            mat[i][j] = arr[i * (i + 1) / 2 + j];
                        else
                            mat[i][j] = 0;
                    }
                }

                cout << "\nLower Triangular Matrix formed from the array is: " << endl;
                for (int x = 0; x < e; x++)
                {
                    for (int y = 0; y < e; y++)
                    {
                        cout << mat[x][y] << "\t";
                    }
                    cout << endl;
                }
            }
            else
            {
                cout << "\n!!! Elements in the Array are more/less than the elements required to form an upper triangular matrix !!!" << endl;
            }
        }
        break;

        case 's':
        {
            int e = (-1 + sqrt(1 + 8 * size)) / 2;
```

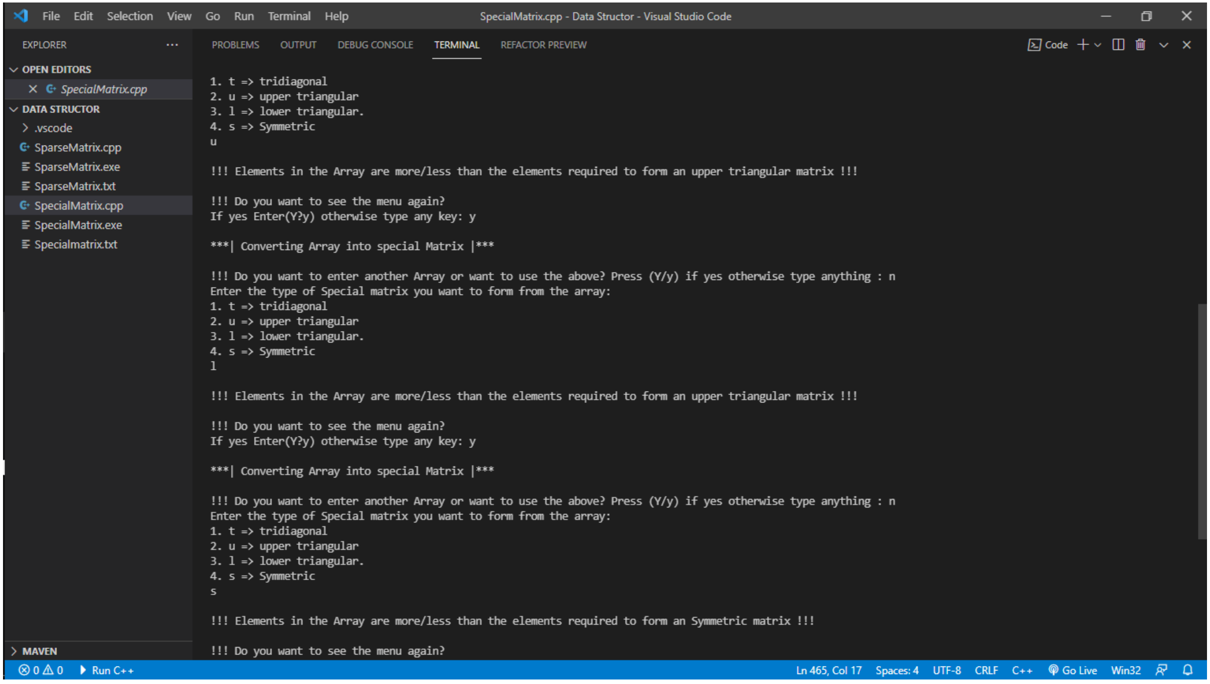
```
if (size == (e * (e + 1) / 2))
{
    int mat[e][e] = {0};

    //converting the array into Symmetric Matrix
    for (int i = 0; i < e; i++)
    {
        for (int j = 0; j < e; j++)
        {
            if (i <= j)
            {
                mat[i][j] = arr[(n * i) - i * (i + 1) / 2 + j];
            }
            else
            {
                mat[i][j] = mat[j][i];
            }
        }
    }

    cout << "\nSymmetric Matrix formed from the array is: " << endl;
    for (int x = 0; x < e; x++)
    {
        for (int y = 0; y < e; y++)
        {
            cout << mat[x][y] << "\t";
        }
        cout << endl;
    }
}
```

```
            else
            {
                cout << "\n!!! Elements in the Array are more/less than the elements required to form an Symmetric matrix !!!" << endl;
            }
        }
        break;

        default:
            cout << "\n!!!Invalid input. Try Again!!!\n";
        }

        cout << "\n!!! Do you want to see the menu again?\nIf yes Enter(Y?y) otherwise type any key: ";
        cin >> selc;
    } while (selc == 'Y' || selc == 'y');

    cout << "\n|*** You choose to Exit the Menu ***|" << endl;
}

int main()
{
    cout << "\n|***|Program Started|***|" << endl;

    matrix obj;
    obj.input();
    obj.checkIfSpecial();
    obj.arrayToMatrix();

    cout << "\n|***|Program Ended|***|" << endl;

    return 0;
```

```
}
```


File Edit Selection View Go Run Terminal Help
SpecialMatrix.cpp - Data Structor - Visual Studio Code
EXPLORER
OPEN EDITORS
SpecialMatrix.cpp
DATA STRUCTOR
.vscode
SparseMatrix.cpp
SparseMatrix.exe
SparseMatrix.txt
SpecialMatrix.cpp
SpecialMatrix.exe
Specialmatrix.txt
MAVEN
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL REFACTOR PREVIEW
Code
0 0 Run C++
Ln 465, Col 17 Spaces: 4 UTF-8 CRLF C++ Go Live Win32

```
PS C:\Users\HP\Desktop\PARENT FOLDER\Data Structor> cd "c:\Users\HP\Desktop\PARENT FOLDER\Data Structor\" ; if ($?) { g++ SpecialMatrix.cpp -o SpecialMatrix }
 ; if ($?) { .\SpecialMatrix }

|***|Program Started|***|
>>>Given Matrix is:
1       0       3       4
0       2       5       8
3       5       6       1
4       0       0       7

<||> The given matrix is not a Special Matrix

***| Converting Array into special Matrix |***

!!! Do you want to enter another Array or want to use the above? Press (Y/y) if yes otherwise type anything : u
Enter the type of Special matrix you want to form from the array:
1. t => tridiagonal
2. u => upper triangular
3. l => lower triangular.
4. s => Symmetric
t

Tridiagonal Matrix formed from the array is:
-858993460      1548236287
458873079       -389231412

!!! Do you want to see the menu again?
If yes Enter(Y?y) otherwise type any key: y

***| Converting Array into special Matrix |***

!!! Do you want to enter another Array or want to use the above? Press (Y/y) if yes otherwise type anything : n
Enter the type of Special matrix you want to form from the array:
1. t => tridiagonal
2. u => upper triangular
3. l => lower triangular.
4. s => Symmetric
u
```


File Edit Selection View Go Run Terminal Help
SpecialMatrix.cpp - Data Structor - Visual Studio Code
EXPLORER
OPEN EDITORS
SpecialMatrix.cpp
DATA STRUCTOR
.vscode
SparseMatrix.cpp
SparseMatrix.exe
SparseMatrix.txt
SpecialMatrix.cpp
SpecialMatrix.exe
Specialmatrix.txt
MAVEN
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL REFACTOR PREVIEW
Code
0 0 Run C++
Ln 465, Col 17 Spaces: 4 UTF-8 CRLF C++ Go Live Win32

```
1. t => tridiagonal
2. u => upper triangular
3. l => lower triangular.
4. s => Symmetric
u

!!! Elements in the Array are more/less than the elements required to form an upper triangular matrix !!!

!!! Do you want to see the menu again?
If yes Enter(Y?y) otherwise type any key: y

***| Converting Array into special Matrix |***

!!! Do you want to enter another Array or want to use the above? Press (Y/y) if yes otherwise type anything : n
Enter the type of Special matrix you want to form from the array:
1. t => tridiagonal
2. u => upper triangular
3. l => lower triangular.
4. s => Symmetric
l

!!! Elements in the Array are more/less than the elements required to form an upper triangular matrix !!!

!!! Do you want to see the menu again?
If yes Enter(Y?y) otherwise type any key: y

***| Converting Array into special Matrix |***

!!! Do you want to enter another Array or want to use the above? Press (Y/y) if yes otherwise type anything : n
Enter the type of Special matrix you want to form from the array:
1. t => tridiagonal
2. u => upper triangular
3. l => lower triangular.
4. s => Symmetric
s

!!! Elements in the Array are more/less than the elements required to form an Symmetric matrix !!!

!!! Do you want to see the menu again?
```