Assignment No.:	4
Title:	Storing of elements in the matrix
Subject:	Data Structures Laboratory.
Class:	S.Y. 13 (BTech) C.S.E.
Roll No.:	2215055 (MIT U21 BT IT0010)
Assessment (Marks):	
Signature and Date of Assessment:	

Experiment No:- 4

* Experiment Title → Implement C++ program for storing matrix.

* Objectives:-

a) To understand the matrix locations for storing the elements.

* Problem Statement:- Write C++ program for storing matrix. Write functions for

i) Check whether given matrix is upper triangular or not.

ii) Compute summation of diagonal elements.

iii) Compute transpose of matrix.

* Outcomes:- Understanding the matrix locations to store elements in different formats.

* Concept:- Upper Triangular Matrix:-

Upper triangular matrix is a square matrix in which all the elements below the principle diagonal are zero. To find the upper triangular matrix, a matrix needs to be a square matrix that is, the number of rows and columns in the matrix needs to be equal. Dimensions of a typical square matrix can be represented by $n \times n$.

$$\begin{bmatrix} 1 & 2 & 3 \\ 8 & 6 & 4 \\ 4 & 5 & 6 \end{bmatrix} \longrightarrow \begin{bmatrix} 1 & 2 & 3 \\ 0 & 6 & 4 \\ 0 & 0 & 6 \end{bmatrix}$$

* Algorithm (To check matrix is upper triangular or not).

1) Start.
2) Declare and initialise a two-dimensional array a.
3) Calculate the number of rows and column present in the array and store it in variables rows and columns respectively.
4) If the number of rows is not equal to the number of columns, it implies that the given matrix is not a square matrix. Hence, given matrix cannot be converted to the upper triangular matrix. Display the error message.
5) If rows = cols, traverse the array a using two loops where outer loop represents the rows, and inner loop represents the column of the array a. To convert given matrix to upper triangular matrix set the elements of the array to 0 where $(i > j)$ that is, the row number is greater than column number.
6) Display the resulting matrix.
7) End.

* Concept :- Summation of diagonal elements :-

→ A square matrix has two diagonals i.e. left and right diagonal. The left diagonal elements have equal row and column indexes i.e. i == j and the sum of indexes of the right diagonal elements is one less than the size of the matrix i.e. i+j == size - 1. Using these relations we can easily get the diagonals elements and find their sum.

1	2	3	4	5
1	2	1	2	3
2	3	4	5	2
1	3	5	1	1
1	2	1	2	1

Sum of left Diagonal : 9
Sum of Right Diagonal : 15

* Algorithm (To compute Summation of diagonal element

1) ~~Start~~
2) ~~Declare a~~

→ Concept ; Transpose of Matrix.

→ A transpose of a matrix is obtained by interchanging rows of the original matrix with columns and vice-versa.

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}^T \longrightarrow \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$$

* Algorithm (To compute summation of diagonal elements)

1) Start
2) Declare the all necessary variables.
3) Enter the order of matrix.
4) Enter the elements of matrix row-wise using loop.
5) Display the entered matrix in standard format (it is not a compulsory step).
6) Assign number of rows with number of column.
7) Swap (i,j) th element with (j,i) th.
8) Store the new elements as ~~elements~~ element of transposed matrix.
9) print the elements of transpose matrix in format wing loop.
10) Stop.

* Conclusion → Thus, we implemented various operations for storing of elements in the matrix.

* Question → What is row major and column major matrix?

→ Row major order and column major order are methods for storing multidimensional arrays in linear storage such as random access memory.

problem statement

Write C++ program forstoring matrix. Write functions for

i. Check whether given matrix is upper triangular or not

ii. Compute summation of diagonal elements

iii. Compute transpose of matrix

Author: Anurag Haldey.

Date:15/09/2022.

**/

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

class matrix
{
 public: int a[20][20],i,j,r,c,isUpper,m,sum=0,sum1=0,sum2=0;
 void matt();
 void show();
 void transpose();
 void upper();
 void diagonal();
};

void matrix::matt()
{
  cout<<"enter rows and columns in matrix\n";
  cin>>r>>c;
  cout<<"enter "<<r*c<<" elements in first matrix\n";
 for(i=0;i<r;i++)
```

```
    {
      for(j=0;j<c;j++)
      {
        cin>>a[i][j];
      }
    }

  }
  void matrix::show()
  {
     cout<<"given matrices are\n\n";
    cout<<"a=";
    for(i=0;i<r;i++)
    {
      for(j=0;j<c;j++)
      {
        cout<<"\t"<<a[i][j];
      }
      cout<<"\n";
    }

      cout<<"\n";

}
void matrix::transpose()
    {
        for(i=0;i<r;i++)
        {
          for(j=0;j>c;j++)
          {
```

```
                cout<<"Transpose of matrix is = ";
                cout<<"\t"<<a[j][i];
              }
              cout<<"\n";
            }
        }

void matrix::upper()
{
 for(i=1;i<r;i++)
 {
   for(j=0;j<i;j++)
   {
     if(a[i][j]!=0)
     {
       isUpper=0;
     }
     else{
       isUpper=1;
     }
   }
}
if(isUpper==1)
{
   cout<<"This is upper Triangular
Matrix."; }
else{
   cout<<"This is not upper triangular
matrix"; }
}
```

```
void matrix::diagonal()
{
   for(i=0;i<r;i++)
   {
      for(j=0;j<c;j++)
      {
         if(i==j)
            sum=sum+a[i][j];
       }
     }
      cout<<"\nsum of diagonal =
   "<<sum; }

   int main()
   {
      matrix m;
      m.matt();
      m.show();
      m.transpose();
      m.upper();
      m.diagonal();

      return 0;
   }
```

Output:

```
enter rows and columns in
matrix 3
3
```

enter 9 elements in first matrix

1

2

3

0

5

8

0

0

7

given matrices are

a= 1 2 3

0 5 8

0 0 7

This is upper Triangular Matrix.

sum of diagonal = 13

Process returned 0 (0x0) execution time : 12.172

s Press any key to continue.

/**

ASSIGNMENT NO:- 5

problem statement- A matrix of m x n order is said to have a saddle point; if some entry at a[i][j] is the smallest value in row i and the largest value in j. Write C++ function that determines the location of a saddle point if it exists.

Author: Anurag Haldey.

Date: 22/09/2022.