



MATRICES

SPECIAL MATRICES

↳ Only square matrix
n x n

1. Diagonal Matrix
2. Lower Triangular
3. Upper Triangular
4. Symmetric Matrix
5. Tridiagonal Matrix
6. Band Matrix
7. Toeplitz Matrix
8. Sparse Matrix

1. DIAGONAL MATRIX

	1	2	3	4	5
1	3	0	0	0	0
2	0	7	0	0	0
3	0	0	4	0	0
4	0	0	0	9	0
5	0	0	0	0	6

A	3	7	4	9	6
	0	1	2	3	4

```
int A[5];  
void set ( int A[], int i, int j, int x )  
{  
    if ( i == j )  
        A[i-1] = x;  
}  
void get ( int A[], int i, int j )  
{  
    if ( i == j )  
        return A[i-1];  
    else  
        return 0;  
}
```

Annotations for the code:

- ROWS: points to the `i` parameter in the `set` function.
- COLUMNS: points to the `j` parameter in the `set` function.
- ELEMENT TO BE INSERTED: points to the `x` parameter in the `set` function.

C++ Class For Diagonal Matrix

```
class Diagonal()
{
```

```
private :
```

```
int n;
int *A;
```

```
public :
```

```
Diagonal(int n)
{
```

```
    This → n = n;
    A = new int(n);
```

```
}
```

```
void set(int i, int j, int k);
```

```
void get(int i, int j);
```

```
void Display();
```

```
~ Diagonal()
```

```
{
```

```
    delete []A;
```

```
}
```

```
}
```

```
void Diagonal :: set(int i, int j, int x);
{
```

```
    if (i == j)
```

```
        A[i-1] = x;
```

```
}
```

```
int Diagonal :: get(int i, int j)
{
```

```
    if (i == j)
```

```
        return A[i-1];
```

```
    else
```

```
        return 0;
```

```
}
```

```
void Diagonal :: Display()
{
```

```
    for (i = 0; i < n; i++)
    {
```

```
        for (j = 0; j < n; j++)
        {
```

```
            if (i == j)
```

```
                cout << A[i-1];
```

```
            else
```

```
                cout << "0";
```

```
        }
```

```
        cout << endl;
```

```
    }
```

```
}
```

LOWER TRIANGULAR MATRIX

$$M = \begin{bmatrix} a_{11} & 0 & 0 & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 & 0 \\ a_{31} & a_{32} & a_{33} & 0 & 0 \\ a_{41} & a_{42} & a_{43} & a_{44} & 0 \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} \end{bmatrix}$$

$$M[i, j] = 0 \quad \text{if } i < j$$

$$M[i, j] = \text{Non Zero} \quad \text{if } i \geq j$$

$$\begin{aligned} \text{Non Zero} &= 1 + 2 + 3 + 4 + 5 \\ &= 1 + 2 + 3 + 4 + \dots + n \\ &= \frac{n(n+1)}{2} \end{aligned}$$

$$\text{Zero} = n^2 - \frac{n(n+1)}{2} = \frac{n(n-1)}{2}$$

Row MAJOR

a_{11}	a_{21}	a_{22}	a_{31}	a_{32}	a_{33}	a_{41}	a_{42}	a_{43}	a_{44}	a_{51}	a_{52}	a_{53}	a_{54}	a_{55}
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
row 1	row 2		row 3			row 4				row 5				

$$\text{Index}(A[4][3]) = [1+2+3] + 2 = 8$$

$$\text{Index}(A[5][4]) = [1+2+3+4] + 3 = 13$$

$$\text{Index}(A[i][j]) = \left[\frac{i(i-1)}{2} \right] + j - 1$$

COLUMN MAJOR

a_{11}	a_{21}	a_{31}	a_{41}	a_{51}	a_{22}	a_{32}	a_{42}	a_{52}	a_{33}	a_{43}	a_{53}	a_{44}	a_{54}	a_{55}
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
col 1					col 2				col 3			col 4		col 5

$$\text{Index}(A[4][4]) = [5+4+3] + 0 = 12$$

$$\text{Index}(A[5][4]) = [5+4+3] + 1 = 13$$

$$\text{Index}(A[5][3]) = [5+4] + 2 = 11$$

$$\text{Index}(A[i][j]) = [n + n-1 + n-2 + \dots + n - (j-2)] + (i-j)$$

$$\begin{aligned} &= [n(j-1) - [1+2+3+\dots+j-2]] + (i-j) \\ &= [n(j-1) - \frac{(j-2)(j-1)}{2}] + (i-j) \end{aligned}$$

UPPER TRIANGULAR MATRIX

$$M = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ 0 & a_{22} & a_{23} & a_{24} & a_{25} \\ 0 & 0 & a_{33} & a_{34} & a_{35} \\ 0 & 0 & 0 & a_{44} & a_{45} \\ 0 & 0 & 0 & 0 & a_{55} \end{bmatrix}$$

$$M[i, j] = 0 \quad \text{if } i > j$$

$$M[i, j] = \text{Non Zero} \quad \text{if } i \leq j$$

$$\begin{aligned} \text{Non Zero} &= 5 + 4 + 3 + 2 + 1 \\ &= \frac{n(n+1)}{2} \end{aligned}$$

$$\text{Zero} = n^2 - \frac{n(n+1)}{2} = \frac{n(n-1)}{2}$$

ROW MAJOR

a_{11}	a_{12}	a_{13}	a_{14}	a_{15}	a_{22}	a_{23}	a_{24}	a_{25}	a_{33}	a_{34}	a_{35}	a_{44}	a_{45}	a_{55}	
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
row1					row2					row3			row4		row5

$$\text{INDEX}(A[4][5]) = [5+4+3]+1 = 13$$

$$\begin{aligned} \text{INDEX}(A[i][j]) &= [n+n-1+n-2+\dots+n-(i-2)] + (j-i) \\ &= \left[(i-1)n - \frac{(i-2)(i-1)}{2} \right] + (j-i) \end{aligned}$$

COLUMN MAJOR

a_{11}	a_{12}	a_{22}	a_{13}	a_{23}	a_{33}	a_{14}	a_{24}	a_{34}	a_{44}	a_{15}	a_{25}	a_{35}	a_{45}	a_{55}
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

$$\text{INDEX}(A[4][5]) = [1+2+3+4]+3 = 13$$

$$\text{INDEX}(A[i][j]) = [1+2+3+\dots+j-1] + i-1 = \left[\frac{j(j-1)}{2} \right] + i-1$$

SYMMETRIC MATRIX

$$M = \begin{bmatrix} 2 & 2 & 2 & 2 & 2 \\ 2 & 3 & 3 & 3 & 3 \\ 2 & 3 & 4 & 4 & 4 \\ 2 & 3 & 4 & 5 & 5 \\ 2 & 3 & 4 & 5 & 6 \end{bmatrix}$$

$$\text{if } M[i, j] = M[j, i]$$

Either we can store lower triangular matrix, or we can store upper triangular matrix

TRI DIAGONAL MATRIX

$$M = \begin{bmatrix} a_{11} & a_{12} & 0 & 0 & 0 \\ a_{21} & a_{22} & a_{23} & 0 & 0 \\ 0 & a_{32} & a_{33} & a_{34} & 0 \\ 0 & 0 & a_{43} & a_{44} & a_{45} \\ 0 & 0 & 0 & a_{54} & a_{55} \end{bmatrix}$$

a_{21}	a_{32}	a_{43}	a_{54}	a_{11}	a_{22}	a_{33}	a_{44}	a_{55}	a_{12}	a_{23}	a_{34}	a_{45}
0	1	2	3	4	5	6	7	8	9	10	11	12
lower diagonal				main diagonal				upper diagonal				

Main diagonal $i - j = 0$
Lower diagonal $i - j = 1$
Upper diagonal $i - j = -1$

$$|i - j| \leq 1$$

$M[i, j] = \text{Non Zero}$ if $|i - j| \leq 1$
 $M[i, j] = \text{Zero}$ if $|i - j| > 1$

$$\begin{aligned} &5 + 4 + 4 \\ &n + n - 1 + n - 1 \\ &3n - 2 \end{aligned}$$

Index ($A[i][j]$)

case 1: if $i - j = 1$ index = $i - 1$

case 2: if $i - j = 0$ index = $n - 1 + i - 1$

case 3: if $i - j = -1$ index = $2n - 1 + i - 1$

SQUARE BAND MATRIX (Same as TRI DIAGONAL matrix)

When there are more than one diagonals below the main diagonal and the number of lower and upper diagonal is equal.

TOEPLITZ MATRIX

$M =$

	1	2	3	4	5
1	2	3	4	5	6
2	7	2	3	4	5
3	8	7	2	3	4
4	9	8	7	2	3
5	10	9	8	7	2

$\uparrow n$ $\uparrow n-1$

2	3	4	5	6	7	8	9	10
0	1	2	3	4	5	6	7	8

row column

$$M[i,j] = M[i-1,j-1]$$

No of elements we want to store: $n + n-1$ *

Index $A[i][j]$

case 1: if $i \leq j$ Index = $j-1$

case 2: if $i > j$ Index = $n + i - j - 1$

CREATING A DYNAMICALLY ALLOCATED ARRAY

```
int *A, n;  
printf("Enter dimension");  
scanf("%d", &n);  
A = (int*) malloc (n * sizeof(int));  
A = new int[n]; C++
```