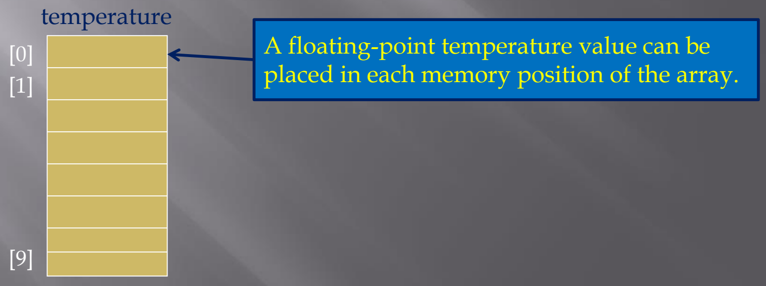
C PROGRAMMING FOR ENGINEERS

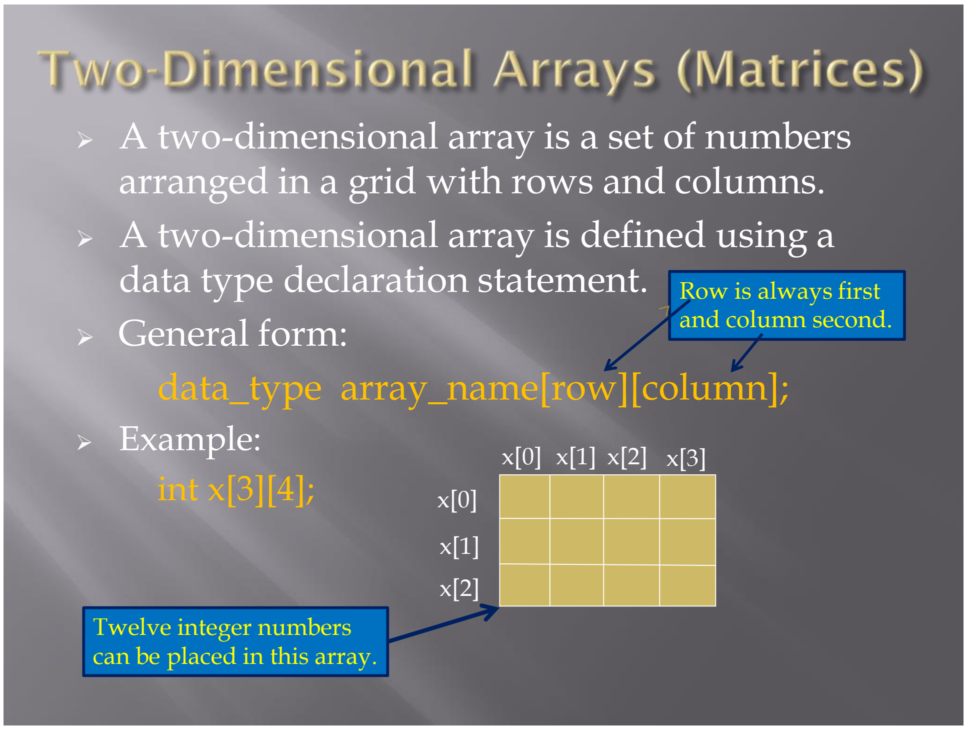
Two-Dimensional Arrays

Review: One-Dimensional Array Example:

double temperature[10];

- ❖ The above declaration allocates memory for 10 floating-point values.
- ❖ Subscript of first element is 0.
- ❖ Subscript of last element is 9.


temperature
[0]
[1]
[9]
A floating-point temperature value can be placed in each memory position of the array.



Two-Dimensional Arrays (Matrices)

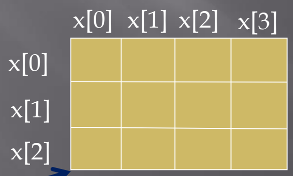
- A two-dimensional array is a set of numbers arranged in a grid with rows and columns.
- A two-dimensional array is defined using a data type declaration statement.
- General form:

  data_type array_name[row][column];

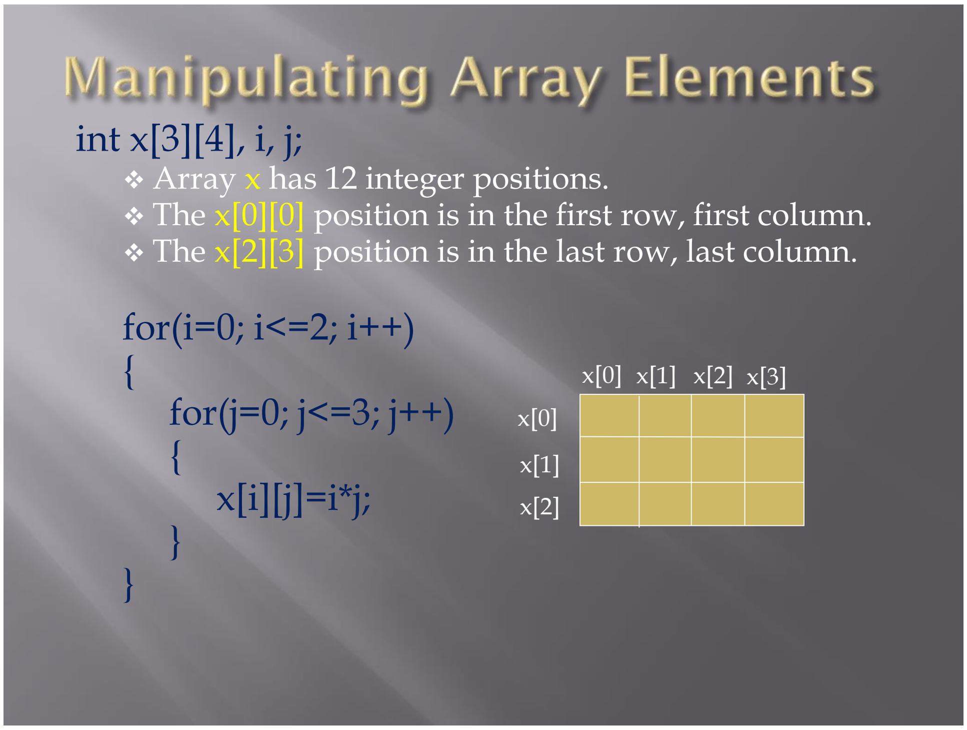
  Row is always first and column second.

- Example:

  int x[3][4];



Twelve integer numbers can be placed in this array.

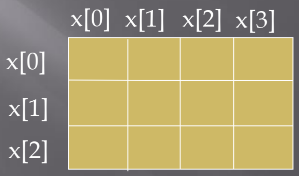


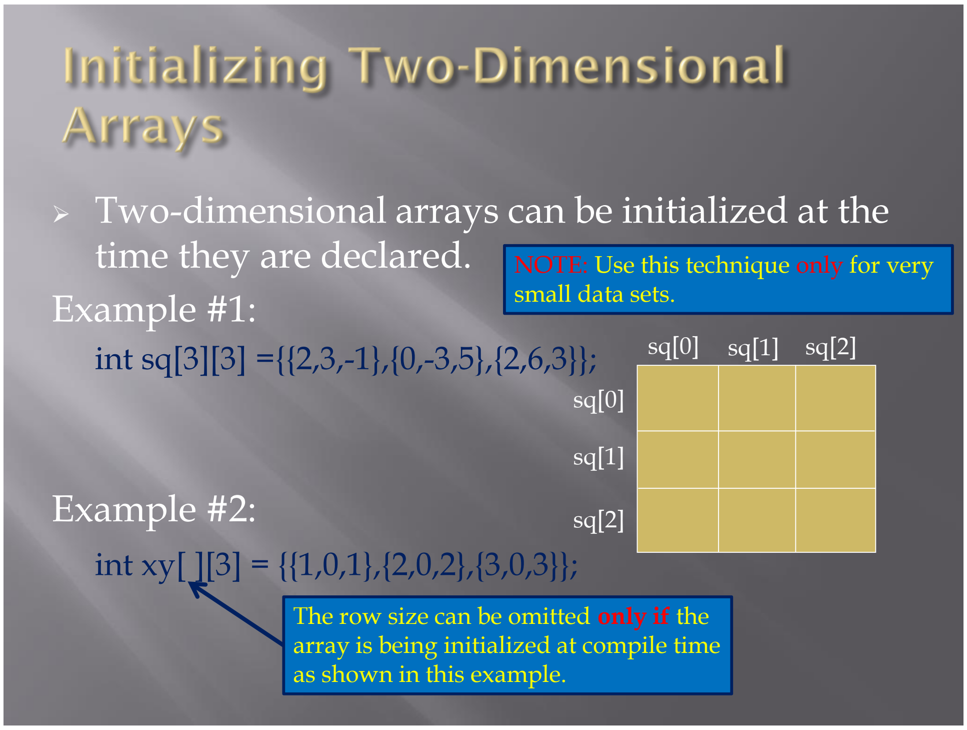
Manipulating Array Elements

int x[3][4], i, j;

- ❖ Array x has 12 integer positions.
- ❖ The x[0][0] position is in the first row, first column.
- ❖ The x[2][3] position is in the last row, last column.

```
for(i=0; i<=2; i++)
{
    for(j=0; j<=3; j++)
    {
        x[i][j]=i*j;
    }
}
```





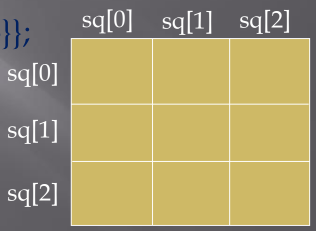
Initializing Two-Dimensional Arrays

- Two-dimensional arrays can be initialized at the time they are declared.

NOTE: Use this technique only for very small data sets.

Example #1:

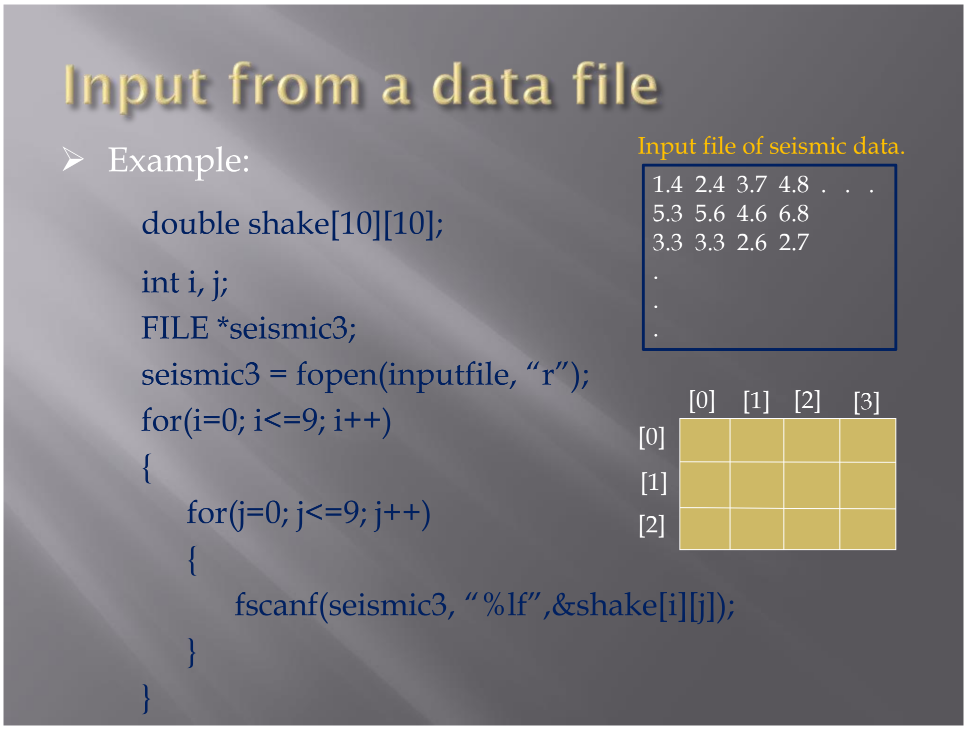
```
int sq[3][3] ={{2,3,-1},{0,-3,5},{2,6,3}};
```



Example #2:

```
int xy[ ][3] = {{1,0,1},{2,0,2},{3,0,3}};
```

The row size can be omitted only if the array is being initialized at compile time as shown in this example.



Input from a data file

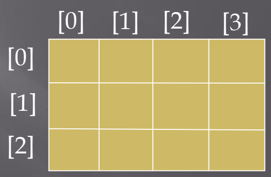
- Example:

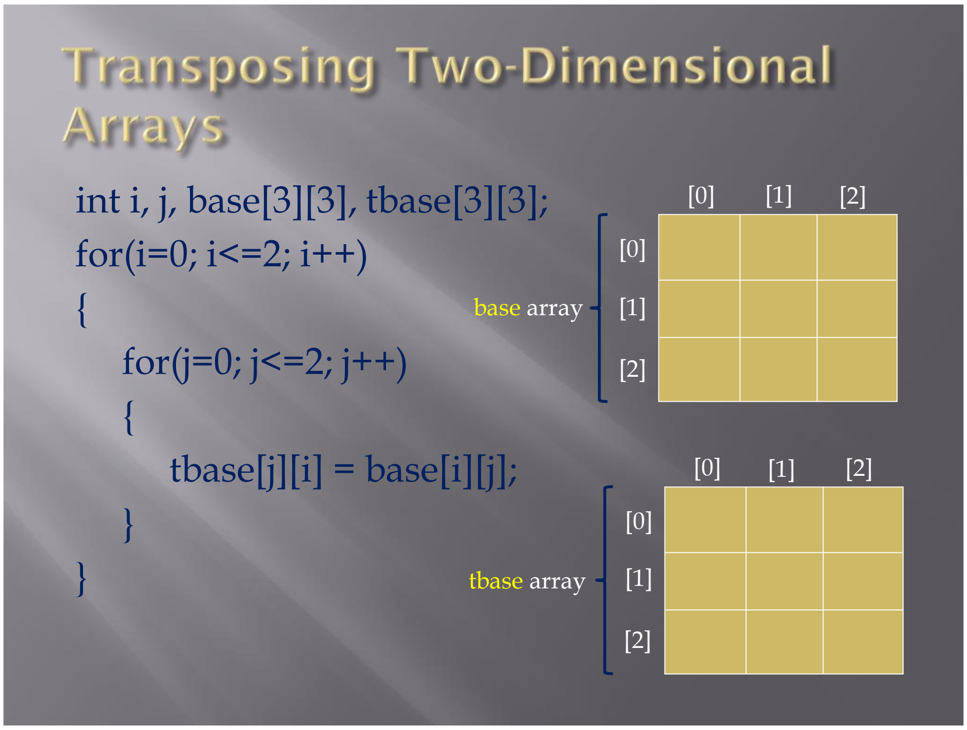
```
double shake[10][10];

int i, j;
FILE *seismic3;
seismic3 = fopen(inputfile, “r”);
for(i=0; i<=9; i++)
{
    for(j=0; j<=9; j++)
    {
        fscanf(seismic3, “%lf”,&shake[i][j]);
    }
}
```

Input file of seismic data.

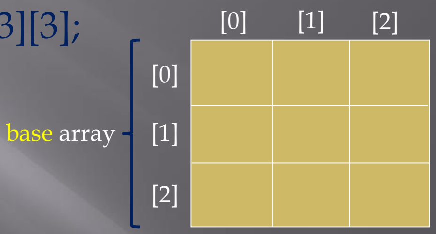
```
1.4  2.4  3.7  4.8  .   .   .
5.3  5.6  4.6  6.8
3.3  3.3  2.6  2.7
.
.
.
```

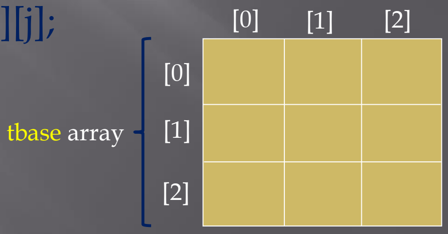




Transposing Two-Dimensional Arrays

```
int i, j, base[3][3], tbase[3][3];
for(i=0; i<=2; i++)
{
    for(j=0; j<=2; j++)
    {
        tbase[j][i] = base[i][j];
    }
}
```


[0]
[1]
[2]
base array
[0]
[1]
[2]


[0]
[1]
[2]
tbase array
[0]
[1]
[2]