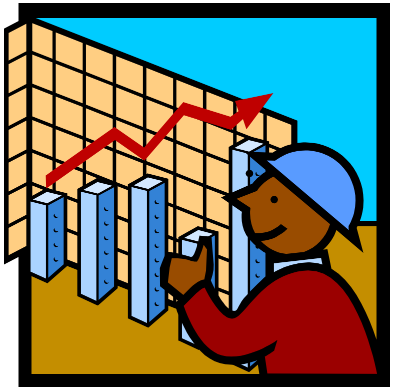
The Matrix

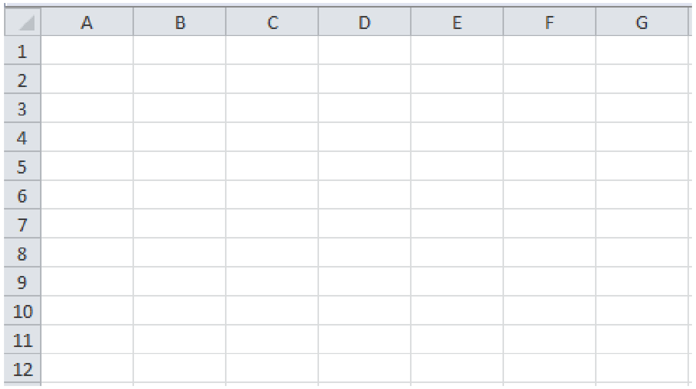
```
1 2 3 4 5 6 7 8 8 1 1 2 3 4 5 6 7 8 8 1
A b a d e a a a a a A b a d e a a a a a
1 4 6 2 a a 3 2 5 1 1 4 6 2 a a 3 2 5 1

1 2 3 4 5 6 7 8 8 1 1 2 3 4 5 6 7 8 8 1
A b a d e a a a a a A b a d e a a a a a
1 4 6 2 a a 3 2 5 1 1 4 6 2 a a 3 2 5 1

1 2 3 4 5 6 7 8 8 1 1 2 3 4 5 6 7 8 8 1
A b a d e a a a a a A b a d e a a a a a
1 4 6 2 a a 3 2 5 1 1 4 6 2 a a 3 2 5 1
```

Two-D arrays
Matrices

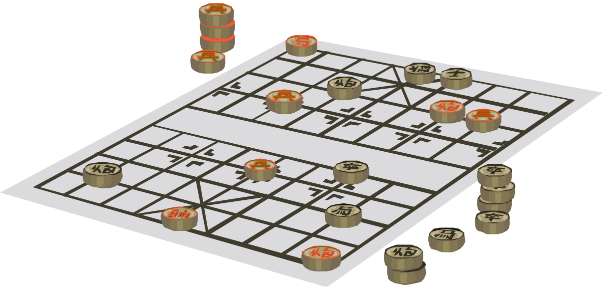
A two-dimensional array is a one-dimensional array of one-dimensional arrays.



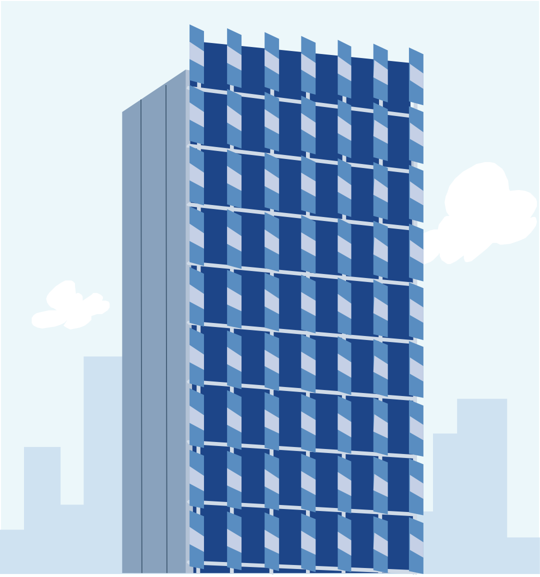
A spreadsheet is a matrix.
A matrix has rows and columns.

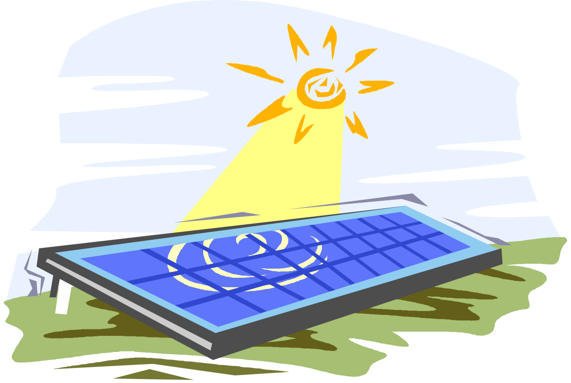


	A	B	C	D	E	F	G
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							



Two-D arrays
Matrices





A solar panel is a large array of solar cells.

What is an array?

An array is a group of items all of the same type which are accessed through a single identifier.

```
int[] nums = new int[10];
```

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

Matrices

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

A matrix is filled with 0 values when instantiated. The exact value in the matrix depends on the specified type.

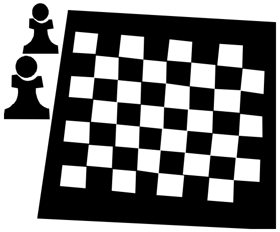
Matrices

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

Each row is a one-dimensional array.

Matrices

	0	1	2	3	4
0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0



```
int[][] mat = new int[5][5];
```

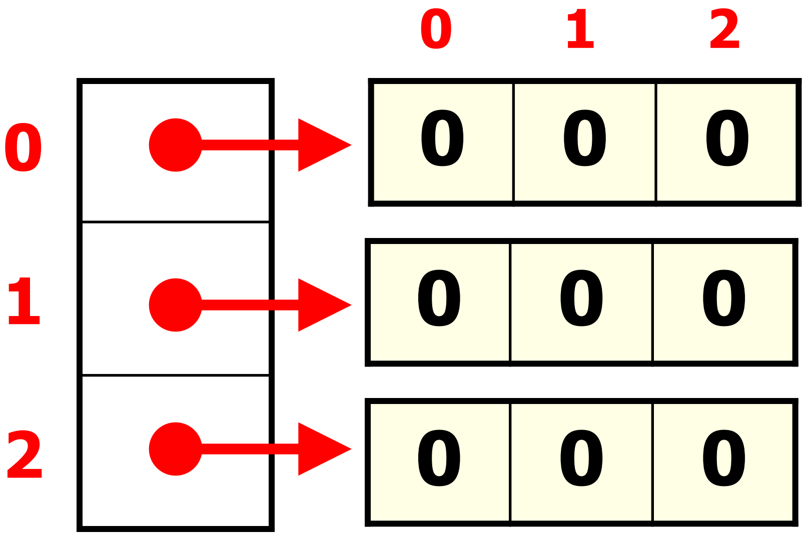
Me & My RC

```
// Row – Column
int[][] mat = new int[3][3];
mat[1][2]=3; // set row 1 col 2 to 3
```

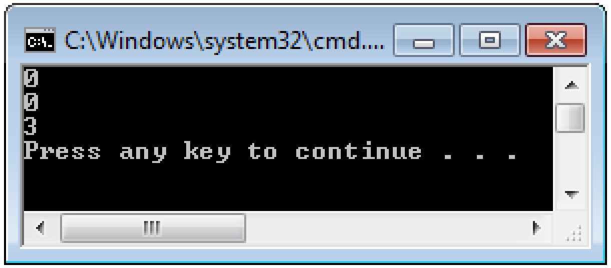
What is a matrix?

A matrix is an array of arrays.

```
int[][] mat = new int[3][3];
```



```
//Matrix Instantiation Example

public class MatrixOne
{
    public static void main(String[] args)
    {
        int[][] mat = new int[3][5];

        System.out.println(mat[2][1]);
        System.out.println(mat[1][2]);
        System.out.println(mat.length);
    }
}
```



```
0
0
3
Press any key to continue . . .
```

Matrix Variables

Matrix Variables

```
// Initializer List
int[][] mat = { {6,5,7,9,2},
                {5,3,4,6,8},
                {7,0,8,5,3} };


final int SIZE = 40;
int[][] intMat = new int[SIZE][SIZE];
//intMat is filled with zeros - 0s
```

Matrix Variables

```
String[][] words = new String[4][4];
//words is filled with 16 nulls

double[][] dMat = new double[3][3];
//dMat is filled with 9 0.0s

int[][] mat = new int[5][5];
//mat is filled with 25 0s
```

Printing Matrix Values

Printing Elements

```
// Jagged arrays are not in the AP Subset...
// but Java lets you do this:
int[][] mat = { {5,7,9,2,1,9},
                {5,3,4},
                {3,7,0,8,9} };

System.out.println(mat[2][1]);
System.out.println(mat[1][2]);
System.out.println(mat[0][3]);
System.out.println(mat[2][4]);
```

OUTPUT

```
7

4

2

9
```

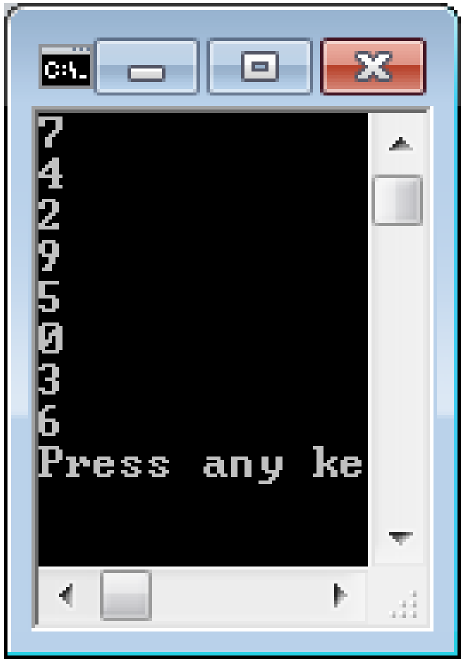
Printing Elements

```
int[][] mat = {{5,7,9,2,1,9},
                {5,3,4},
                {3,7,0,8,9}};

System.out.println(mat[7/4][0]);
System.out.println(mat[1*2][2]);
System.out.println(mat.length);
System.out.println(mat[0].length);
```

OUTPUT

```
5
0
3
6
```

```
//Matrix Instantiation Example

public class MatrixTwo
{
    public static void main(String args[])
    {
        int[][] mat = {{5,7,9,2,1,9},
                       {5,3,4},
                       {3,7,0,8,9}};

        System.out.println(mat[2][1]);
        System.out.println(mat[1][2]);
        System.out.println(mat[0][3]);
        System.out.println(mat[2][4]);
        System.out.println(mat[7/4][0]);
        System.out.println(mat[1*2][2]);
        System.out.println(mat.length);
        System.out.println(mat[0].length);
    }
}
```



```
7
4
2
9
5
0
3
6
Press any ke
```

Setting Matrix Values

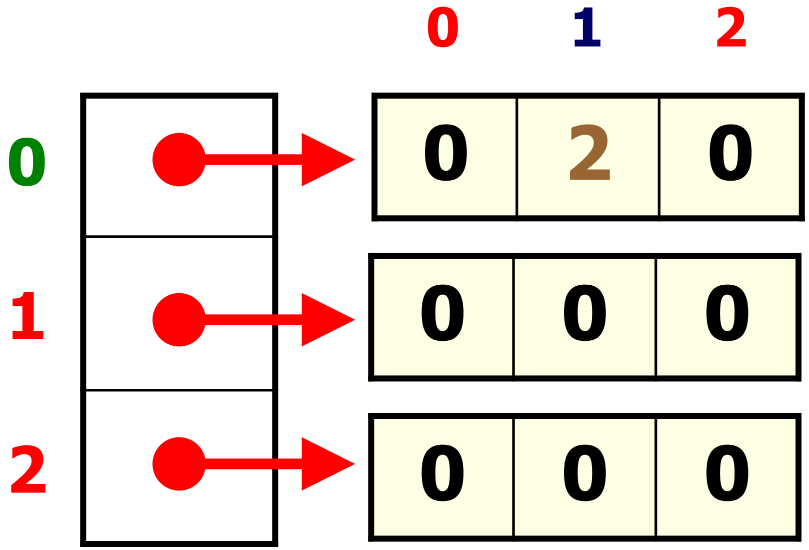
What is a matrix?

A matrix is an array of arrays.

```
int[][] mat = new int[3][3];
mat[0][1]=2;
```

Which array?

Which spot?

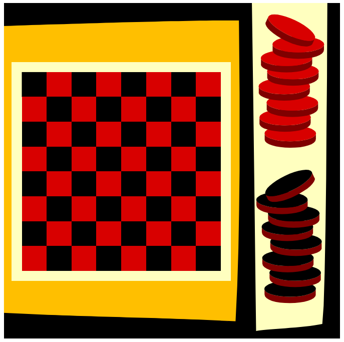


	0	1	2
0	0	2	0
1	0	0	0
2	0	0	0

Assigning Matrix Values

	0	1	2	3	4
0	0	0	0	5	0
1	0	0	0	0	0
2	0	0	7	0	0
3	0	0	0	0	0
4	0	3	0	0	0

```
mat[2][2]=7;

mat[0][3]=5;

mat[4][1]=3
```



Assigning Matrix Values

```
for( int r = 0; r < mat.length; r++)
{
   for( int c = 0; c < mat[r].length; c++)
   {
      mat[r][c] = r*c;
   }
}
```

Rows

Cols

if mat was 3x3:

	0	1	2
0	0	0	0
1	0	1	2
2	0	2	4

Nested Loop Review

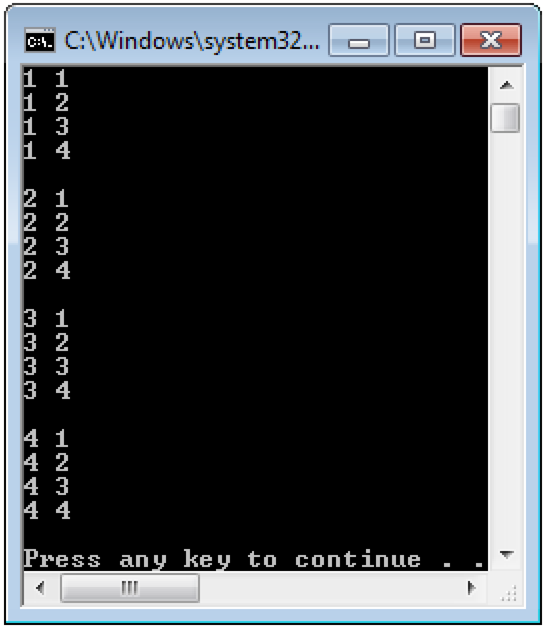
Nested Loop Review

```
int i=1;
       //start  //stop    //increment
for(  i=1;      i<=2;     i++)
{
          //start    //stop  //increment
    for(int j=1;  j<=2;   j++)
        System.out.println(i + " " + j);
    System.out.println();
}
```

OUTPUT

```
1 1
1 2

2 1
2 2
```

```
//Nested For Loop Example

public class NestedFor
{
   public static void main(String[] args)
   {
        int stop = 4;

             //start              //stop              //increment
        for(int outer=1;    outer<=stop;    outer++)
        {
                //start             //stop             //increment
            for(int inner=1;  inner<=stop;    inner++)
                System.out.println(outer + " " + inner);
            System.out.println();
        }
    }
}
```



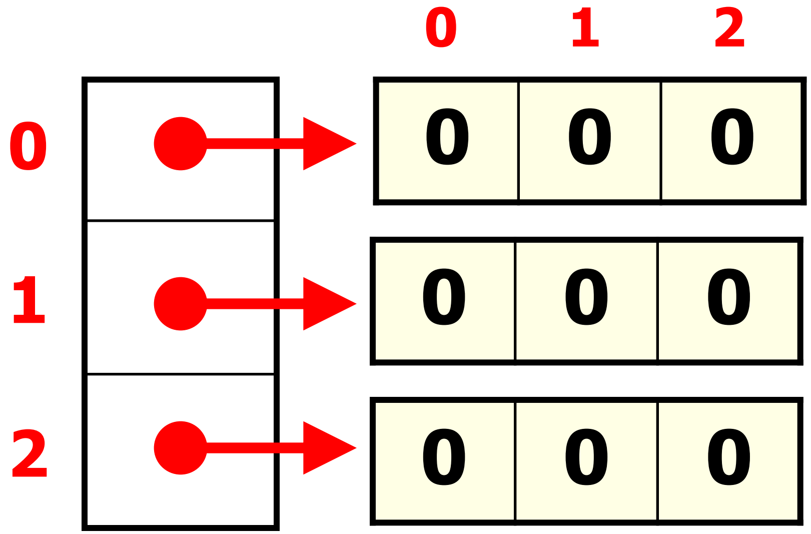
```
1 1
1 2
1 3
1 4

2 1
2 2
2 3
2 4

3 1
3 2
3 3
3 4

4 1
4 2
4 3
4 4

Press any key to continue . .
```

Processing Matrices With Loops

What is a matrix?

A matrix is an array of arrays.

```
int[][] mat = new int[3][3];
```



# of arrays

(rows)

size of each array

(cols)

Printing an Array

```
int[][] mat = { {5,7}, {5,3,4,6}, {0,8,9} };

System.out.println(Arrays.toString(mat[0]));

System.out.println(Arrays.toString(mat[1]));
```

OUTPUT

```
[5, 7]
[5, 3, 4, 6]
```

Printing an Array

```
int[] nums = {1,2,3,4,5,6,7};
for(int r=0; r<nums.length; r++)
{
    System.out.println(nums[r]);
}
```

length returns the # of elements/items/spots in the array!!!

OUTPUT

```
1
2
3
4
5
6
7
```

Printing a row

```
int[][] mat = {{5,7},{5,3,4,6},{0,8,9}};

for(int c=0; c<mat[1].length; c++)
{
   System.out.print(mat[1][c]);
}
System.out.println();
```

OUTPUT

```
5 3 4 6
```

Printing a Matrix

```
int[][] mat = {{5,7},{5,3,4,6},{0,8,9}};
for(int r=0; r<mat.length; r++)
{
   for(int c=0; c<mat[r].length; c++)
   {
      System.out.print(mat[r][c]);
   }
   System.out.println();
}
```

OUTPUT

```
5 7
5 3 4 6
0 8 9
```

Printing a Matrix

```
int[][] mat = {{5,7},{5,3,4,6},{0,8,9}};

for( int[] row : mat )
{
  for( int num : row )
  {
    System.out.print( num + " ");
  }
  System.out.println();
}
```

OUTPUT

```
5 7
5 3 4 6
0 8 9
```

Searching for Values

Searching a Matrix

```
int[][] mat = {{5,7},{5,3,4,6},{0,8,9}};
int count = 0;
for( int r = 0; r < mat.length; r++ )
{
   for( int  c = 0; c < mat[r].length;  c++)
   {
      if( mat[r][c] == 5 )
        count++;
   }
}
System.out.println("5 count = "  + count);
```

OUTPUT

```
5 count = 2
```

Searching a Matrix

```
int[][] mat = {{5,7},{5,3,4,6},{0,8,9}};
int count = 0;
for( int[] row : mat )
{
  for( int num : row )
  {
    if( num == 5 )
      count++;
  }
}
System.out.println("5 count = "  + count);
```

OUTPUT

```
5 count = 2
```

Matrices As Instance Vars

Matrix Instance Vars

```
public class MatrixFun
{
  private int[][] mat;     //instance variable

  public MatrixFun(int numRows, int numCols)
  {
      mat = new int[numRows][numCols];
  }

  public void setSpot(int row, int col, int val)
  {
              mat[row][col] = val;
  }
}
```

Remember...

2D Matrices like RC.

Rows first - - Columns second