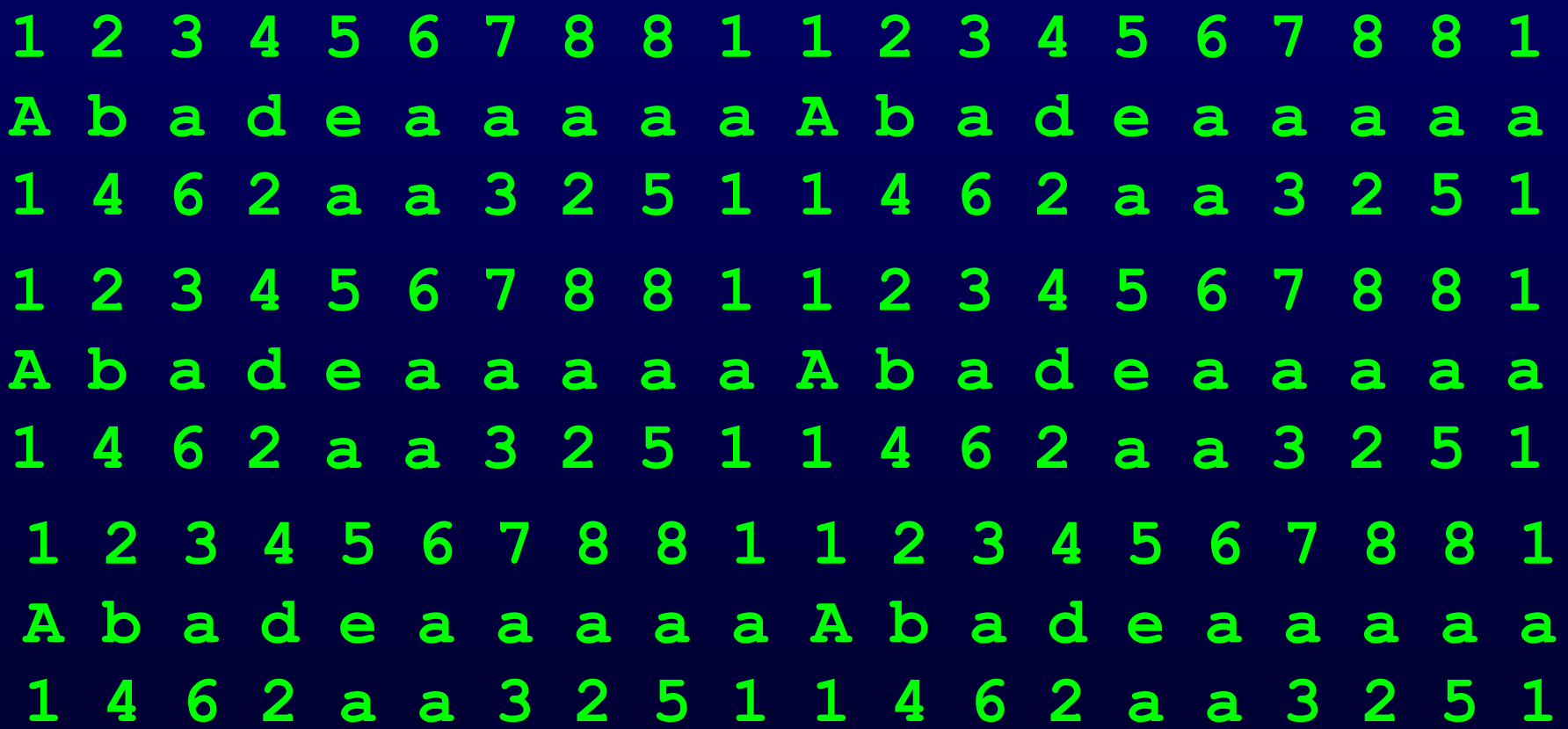
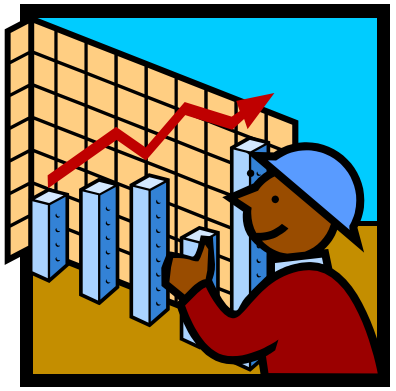
The Matrix



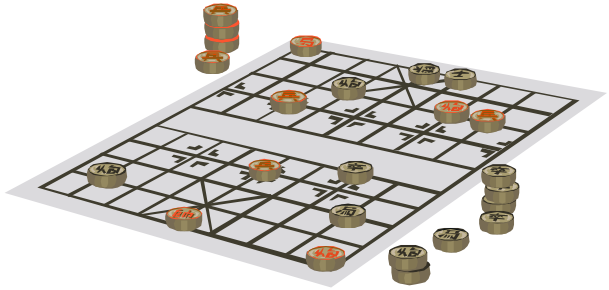
Two-D arrays
Matrices

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

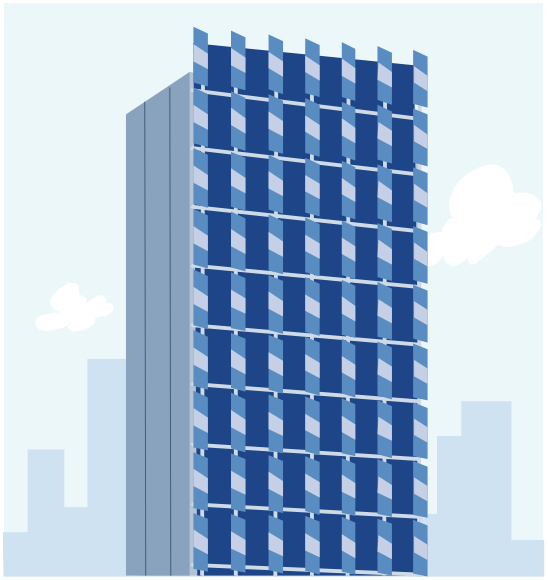


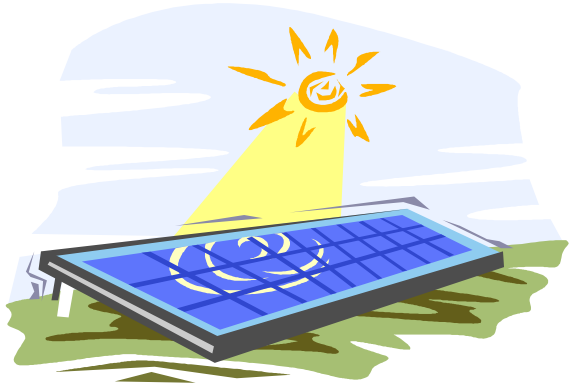
A spreadsheet is a matrix.

A matrix has rows and columns.



Two-D arrays
Matrices



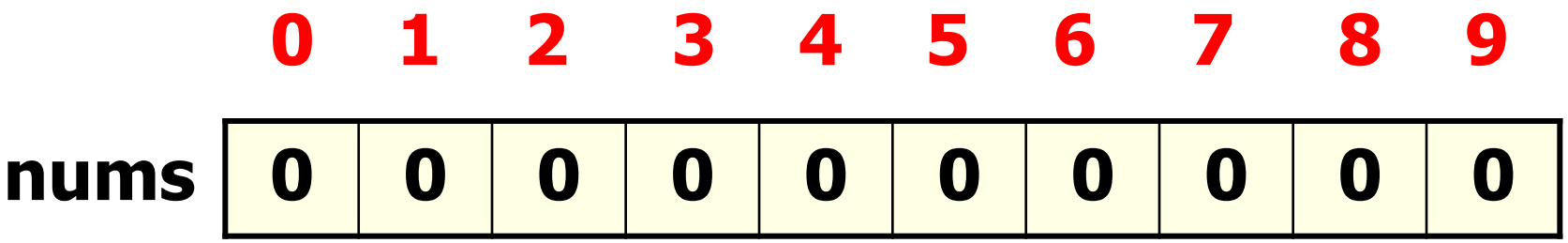


A solar panel is a large array of solar cells.

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];
```



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.

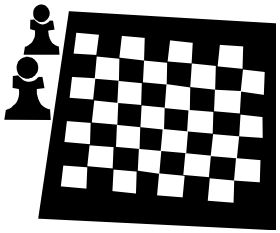
Matrices

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

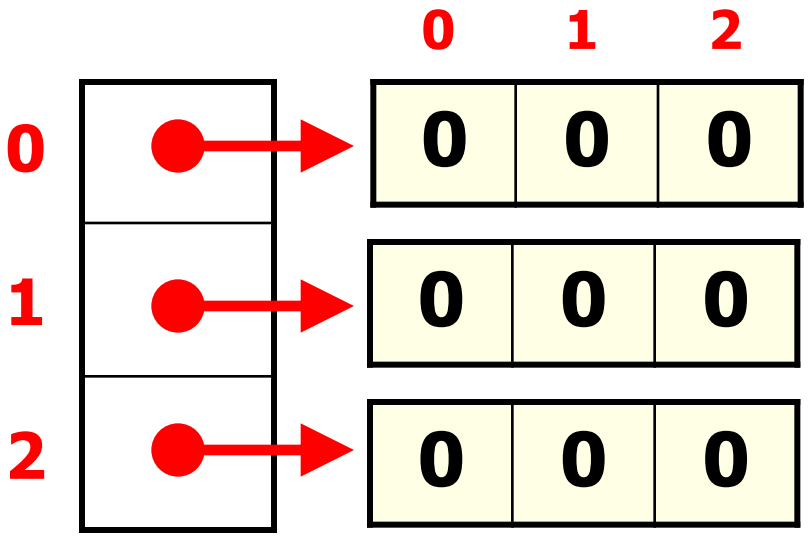


```
int[][] mat = new int[5][5];
```

What is a matrix?

A matrix is an array of arrays.

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



open matrixone.java

Matrix Variables

Matrix Variables

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

```
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 Spots

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

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

OUTPUT

```
7
4
2
9
```

Printing Spots

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

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

OUTPUT

```
5
0
3
6
```

open matrixtwo.java

Setting Matrix Values

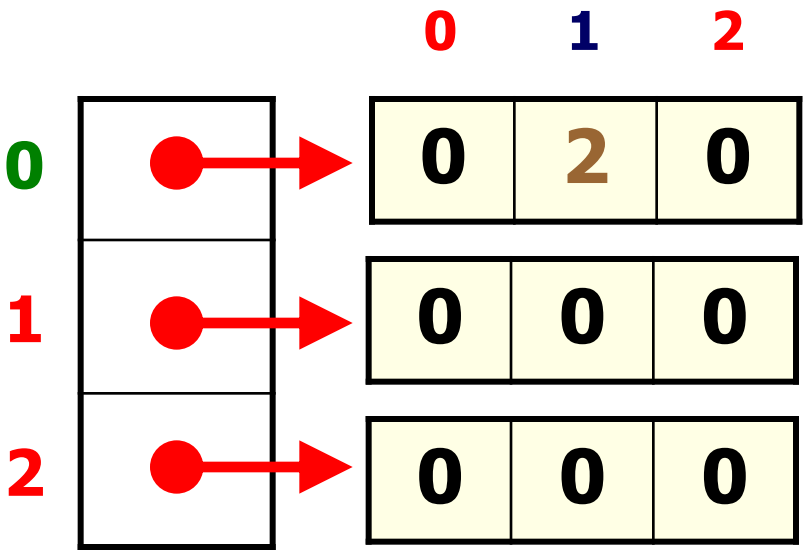
What is a matrix?

A matrix is an array of arrays.

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

Which array?

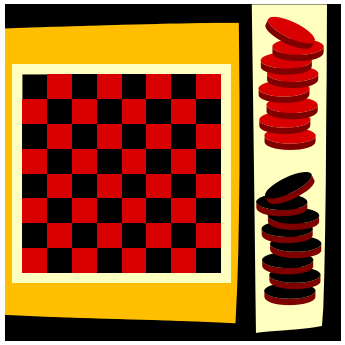
Which spot?



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;
  }
}
```

if mat was 3x3

0	0	0
0	1	2
0	2	4

open matrixsetone.java matrixsettwo.java

Nested Loop Review

Nested Loop Review

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

OUTPUT

```
1 1
1 2

2 1
2 2
```

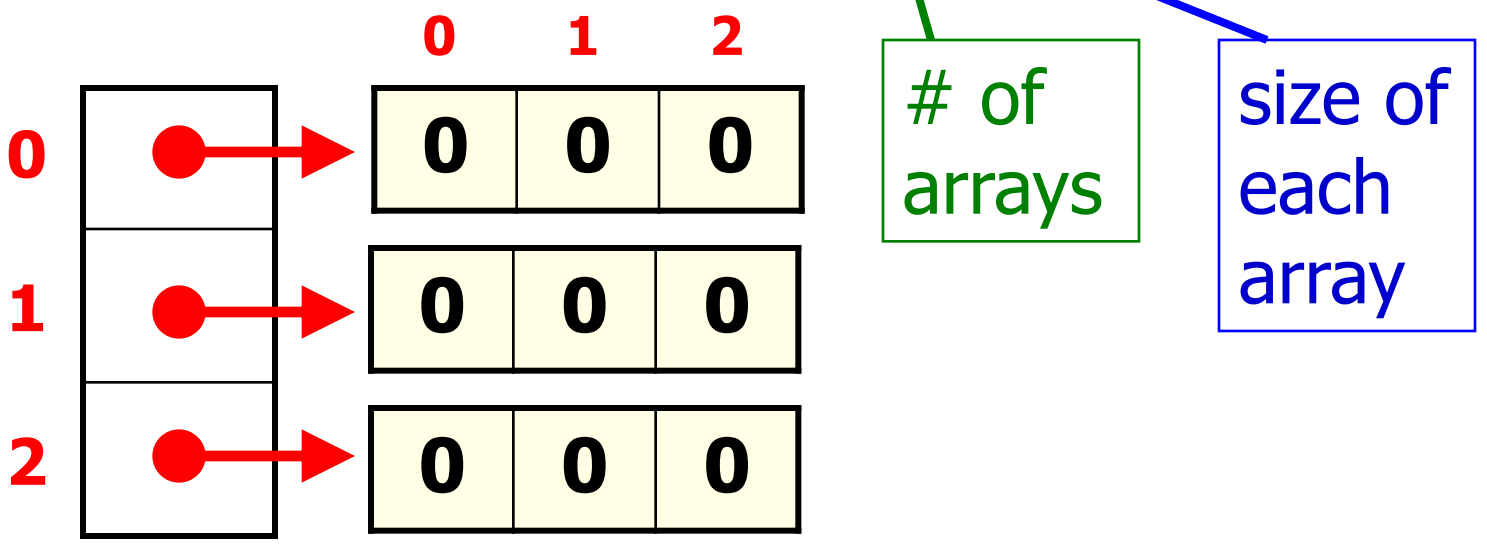
open nestedfor.java

Processing Matrices With Loops

What is a matrix?

A matrix is an array of arrays.

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



Printing an Array

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

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

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++)
{
    out.println(nums[r]);
}
```

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

OUTPUT

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

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++)
    {
        out.print(mat[r][c]);
    }
    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
```

open matrixoutone.java

matrixouttwo.java

open matrixoutthree.java

matrixoutfour.java

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
```

open matrixsearch.java

Summing a Matrix

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

OUTPUT

47

open matrixsum.java

Matrix of References

```
public class Dog
{
  private int age;
  private String name;

  public Dog( String n, int a )  {
    age = a;
    name = n;
  }

  public int getAge()    {
    return age;
  }

  public String getName()    {
     return name;
  }

  public String toString()    {
    return "Dog - " + name + " " + age;
  }
}
```

Basic Dog Class

Matrix of References

```
Dog[][] herd;
herd = new Dog[3][3];

herd[0][0] = new Dog( "fred", 11) ;
herd[1][2] = new Dog( "ann", 21) ;

System.out.println( herd[2][2] );
System.out.println( herd[0][0] );
```

OUTPUT

```
null
Dog - fred 11
```

Open Dog.java

Doggies.java

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];
  }

  //other methods not shown
}
```

Matrix Instance Vars

```
public class Doggies
{
  private Dog[][] mat;   //instance variable

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

  //other methods not shown
}
```

open matrixinstancevars.java

Matrix Extras

matrixinout.java

A complete matrix program.

matrixtotal.java

A matrix program that totals a matrix.

matrixfilereaderfor.java

A matrix program that reads a matrix from a file.

Helpful Hint

2D Matrices like RC.

Rows first - - Columns second

Start work on the labs