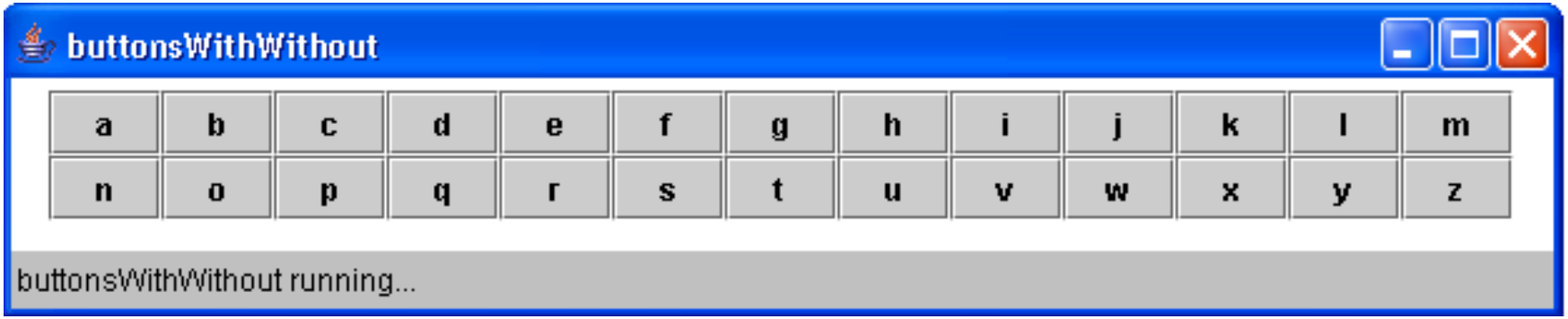
Simple Arrays

Suppose you want this:



Currently, you'd code a bunch of lines like this:

```
Panel p = new Panel (new GridLayout (2, 13));

b0 = new JButton ("a");
b0.setActionCommand ("a");
b0.addActionListener (this);
p.add(b0);

b1 = new JButton ("b");
b1.setActionCommand ("b");
b1.addActionListener (this);
p.add(b1);

Etc.
```

To do the alphabet, you'd need all of this.

```
Panel p = new Panel (new GridLayout (2, 13));

b0 = new JButton ("a");
b0.setActionCommand ("a");
b0.addActionListener (this);
p.add(b0);

b1 = new JButton ("b");
b1.setActionCommand ("b");
b1.addActionListener (this);
p.add(b1);

b2 = new JButton ("c");
b2.setActionCommand ("c");
b2.addActionListener (this);
p.add(b2);

b3 = new JButton ("d");
b3.setActionCommand ("d");
b3.addActionListener (this);
p.add(b3);

b4 = new JButton ("e");
b4.setActionCommand ("e");
b4.addActionListener (this);
p.add(b4);

b5 = new JButton ("f");
b5.setActionCommand ("f");
b5.addActionListener (this);
p.add(b5);

b6 = new JButton ("g");
b6.setActionCommand ("g");
b6.addActionListener (this);
p.add(b6);

b7 = new JButton ("h");
b7.setActionCommand ("h");
b7.addActionListener (this);
p.add(b7);

b8 = new JButton ("i");
b8.setActionCommand ("i");
b8.addActionListener (this);
p.add(b8);

b9 = new JButton ("j");
b9.setActionCommand ("j");
b9.addActionListener (this);
p.add(b9);

b10 = new JButton ("k");
b10.setActionCommand ("k");
b10.addActionListener (this);
p.add(b10);

b11 = new JButton ("l");
b11.setActionCommand ("l");
b11.addActionListener (this);
p.add(b11);

b12 = new JButton ("m");
b12.setActionCommand ("m");
b12.addActionListener (this);
p.add(b12);

b13 = new JButton ("n");
b13.setActionCommand ("n");
b13.addActionListener (this);
p.add(b13);

b14 = new JButton ("o");
b14.setActionCommand ("o");
b14.addActionListener (this);
p.add(b14);

b15 = new JButton ("p");
b15.setActionCommand ("p");
b15.addActionListener (this);
p.add(b15);

b16 = new JButton ("q");
b16.setActionCommand ("q");
b16.addActionListener (this);
p.add(b16);

b17 = new JButton ("r");
b17.setActionCommand ("r");
b17.addActionListener (this);
p.add(b17);

b18 = new JButton ("s");
b18.setActionCommand ("s");
b18.addActionListener (this);
p.add(b18);

b19 = new JButton ("t");
b19.setActionCommand ("t");
b19.addActionListener (this);
p.add(b19);

b20 = new JButton ("u");
b20.setActionCommand ("u");
b20.addActionListener (this);
p.add(b20);

b21 = new JButton ("v");
b21.setActionCommand ("v");
b21.addActionListener (this);
p.add(b21);

b22 = new JButton ("w");
b22.setActionCommand ("w");
b22.addActionListener (this);
p.add(b22);

b23 = new JButton ("x");
b23.setActionCommand ("x");
b23.addActionListener (this);
p.add(b23);

b24 = new JButton ("y");
b24.setActionCommand ("y");
b24.addActionListener (this);
p.add(b24);

b25 = new JButton ("z");
b25.setActionCommand ("z");
b25.addActionListener (this);
p.add(b25);

add (p);
```


133 lines.

Compare that to arrays:

```
JButton b[] = new JButton [26];
Panel p = new Panel (new GridLayout (2, 13));
for (int i = 0 ; i < b.length ; i++)
{
    b [i] = new JButton ((char) (i + 97) + "");
    b [i].setActionCommand ((char) (i + 97) + "");
    b [i].addActionListener (this);
    p.add (b [i]);
}
add (p);
```

Yes, that's all:

```
JButton b[] = new JButton [26];
Panel p = new Panel (new GridLayout (2, 13));
for (int i = 0 ; i < b.length ; i++)
{
    b [i] = new JButton ((char) (i + 97) + "");
    b [i].setActionCommand ((char) (i + 97) + "");
    b [i].addActionListener (this);
    p.add (b [i]);
}
add (p);
```


9 lines.
vs. 133 lines.


The more things,
the more arrays save you time!

Array Declaration:

```
JButton b[] = new JButton [26];
Panel p = new Panel (new GridLayout (2, 13));
for (int i = 0 ; i < b.length ; i++)
{
    b [i] = new JButton ((char) (i + 97) + "");
    b [i].setActionCommand ((char) (i + 97) + "");
    b [i].addActionListener (this);
    p.add (b [i]);
}
add (p);
```

Just [ ] is needed to switch from variable to array:

```
JButton b[] = new JButton [26];
Panel p = new Panel (new GridLayout (2, 13));
for (int i = 0 ; i < b.length ; i++)
{
    b [i] = new JButton ((char) (i + 97) + "");
    b [i].setActionCommand ((char) (i + 97) + "");
    b [i].addActionListener (this);
    p.add (b [i]);
}
add (p);
```

FinalGame

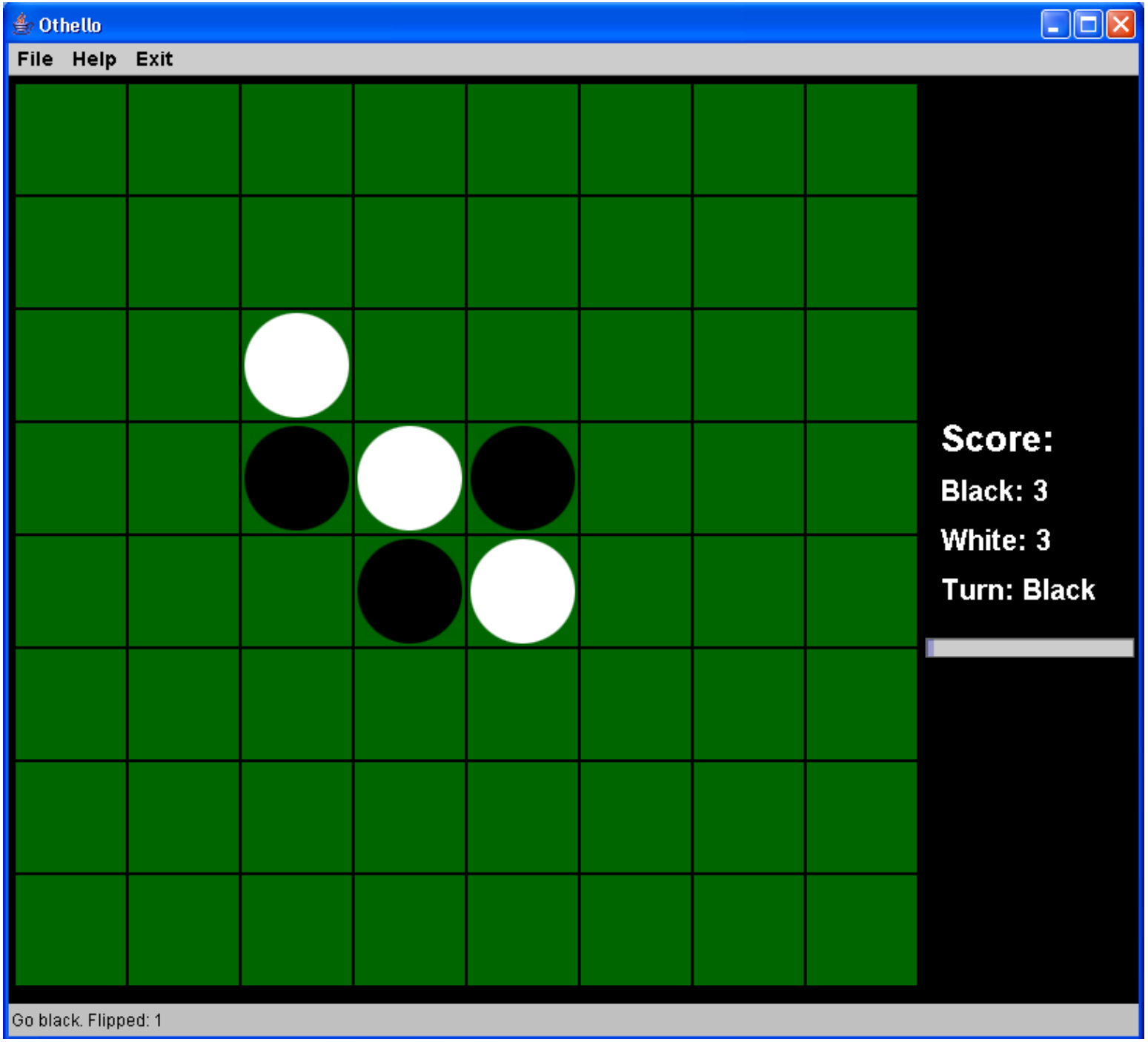
SPACE QUEST

CONNECT 4

Reset | Instructions | Skip Turn

Turn:  Player 1: 0 Player 2: 0

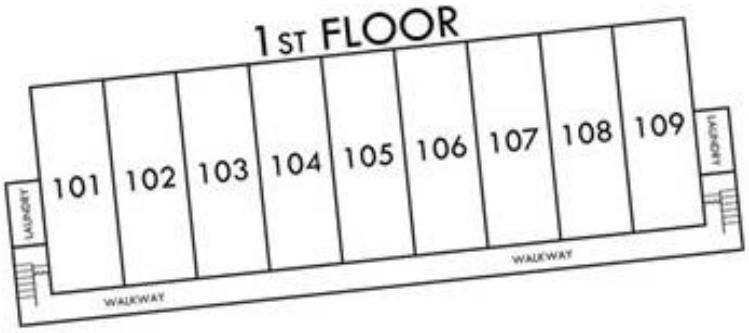
FinalGame running...

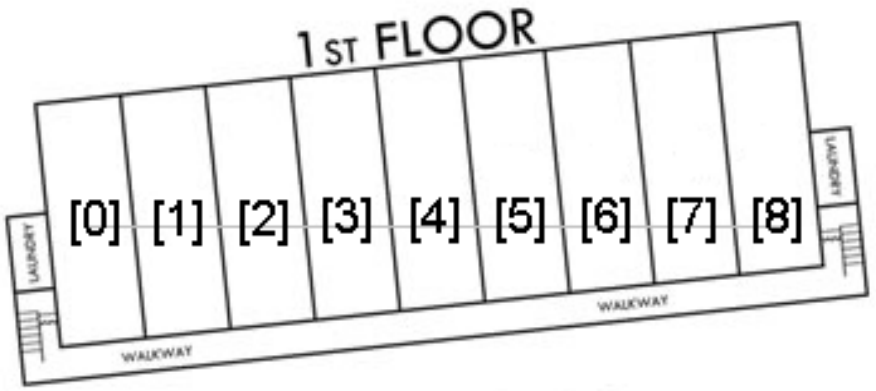

Othello
File
Help
Exit
Score:
Black: 3
White: 3
Turn: Black
Go black. Flipped: 1

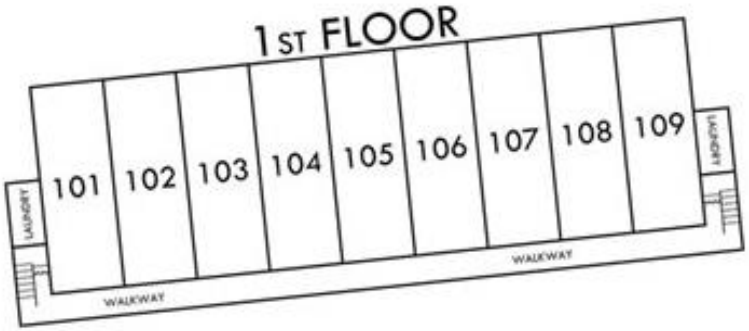

FinalGame
POTIONS 2048
?
Score: 0
FinalGame running...

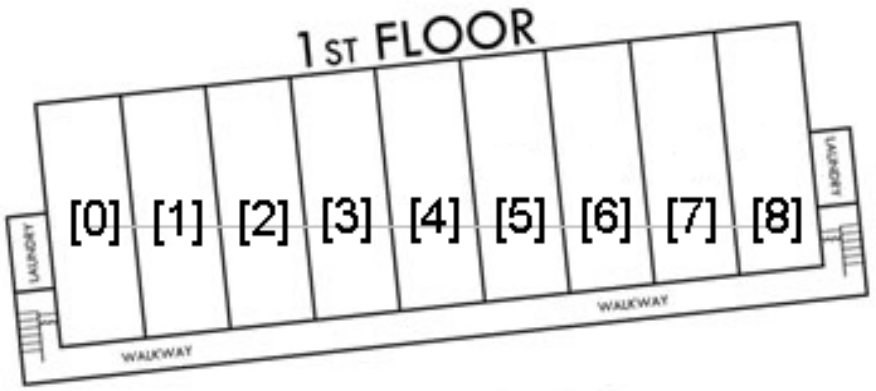
Array

- A group of variables saved under one name.
- They are useful because it is easy to loop through them and change or use all of the variables quickly.
- They allow us to store data. We can sort it or search it if it is in an array.


1ST FLOOR
101
102
103
104
105
106
107
108
109
LAUNDRY
LAUNDRY
WALKWAY
WALKWAY

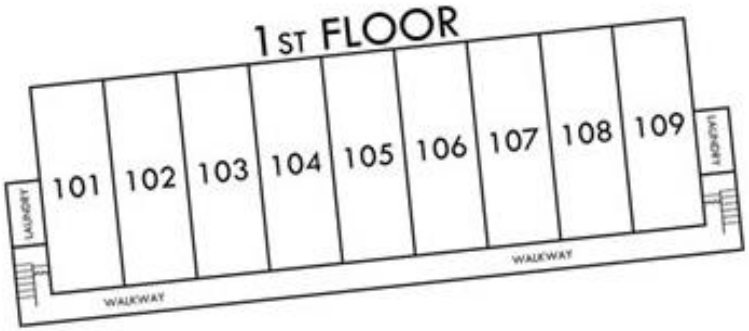

1ST FLOOR
[0]
[1]
[2]
[3]
[4]
[5]
[6]
[7]
[8]
LAUNDRY
LAUNDRY
WALKWAY
WALKWAY

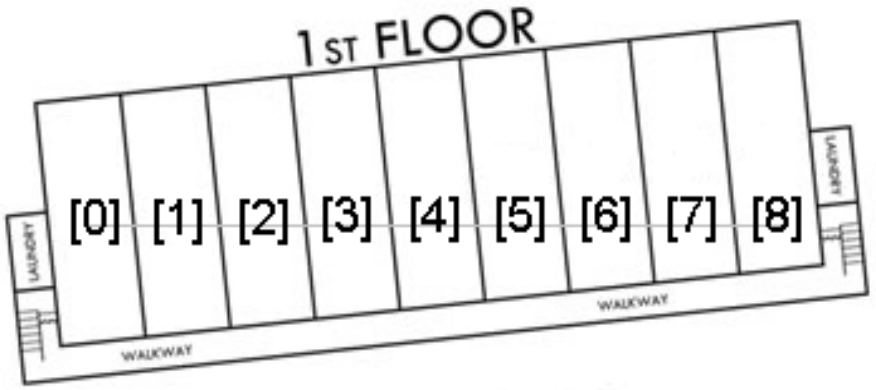




Declaring with data:

```
int apartment [] = {101, 102, 103,
 104, 105, 106, 107, 108, 109};
```





Declaring with data:

```
int apartment [] = {101, 102, 103,
 104, 105, 106, 107, 108, 109};
```

It would make this in RAM:

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]
101	102	103	104	105	106	107	108	109

Index = an address, plural = indices

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]
101	102	103	104	105	106	107	108	109

Index = an address, plural = indices

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]
101	102	103	104	105	106	107	108	109

Element = a place in the array

Index = an address, plural = indices

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]
101	102	103	104	105	106	107	108	109

Element = a place in the array

What is in element 0?

Index = an address, plural = indices

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]
101	102	103	104	105	106	107	108	109

Element = a place in the array

What is in element 0?

101

Index = an address, plural = indices

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]
101	102	103	104	105	106	107	108	109

Element = a place in the array

What is in element 0?

101

What is the index of 106?

Index = an address, plural = indices

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]
101	102	103	104	105	106	107	108	109

Element = a place in the array

What is in element 0?

101

What is the index of 106?

5

Index

- The address of an element.
- The indices go from 0 to the array length - 1.
- Our array loop goes through all of the indices:

```
for(int i=0; i<a.length; i++)
```

Element

- A value stored at a specific index.
- Only one things is stored in each spot.
- All of the things are the same type.
- To find what is stored in an element: name[3]
- To change what is in an elemet: name[3]=567;

Declaring without data:

```
char scrabble [] = new char[7];
```

Adding data later:

```
scrabble [0] = ‘z’;
```

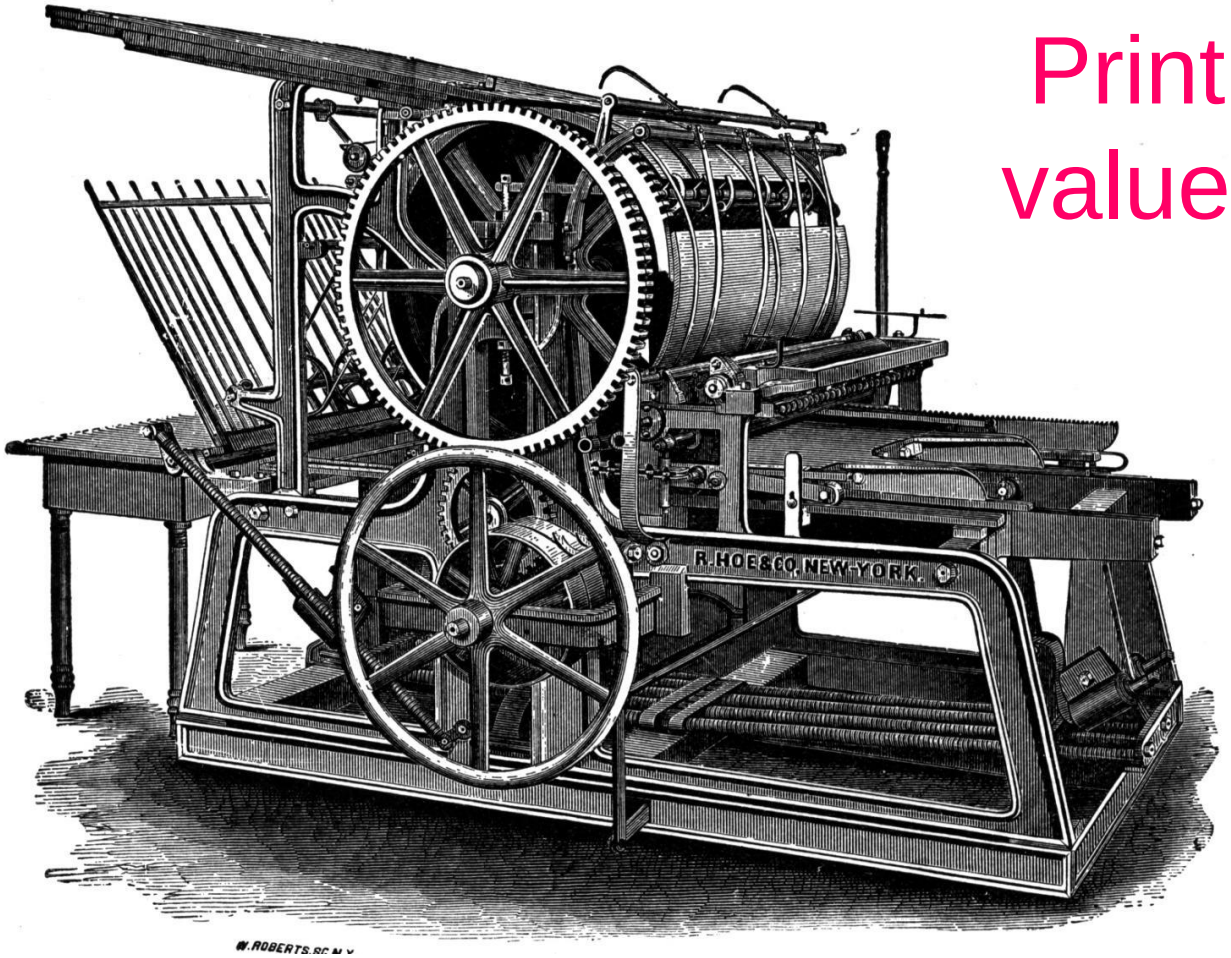
Declaring with data:

```
char scrabble [] = {‘a’, ‘g’,
‘t’, ‘a’, ‘b’, ‘f’, ‘m’};
```

```
String week [ ] = { “Sunday”, “Monday”, “Tuesday”, “Wednesday”, “Thursday”, “Friday”, “Saturday” };
```

week

[0]	[1]	[2]	[3]	[4]	[5]	[6]
Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday



Printing the values in an array

week

[0]	[1]	[2]	[3]	[4]	[5]	[6]
Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday

```
for (int i =0; i < week.length; i++)
    System.out.println(week[i]);
```

Start i at 1

Stop when you get to 7

Move along 1

week

[0]	[1]	[2]	[3]	[4]	[5]	[6]
Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday

```
for (int i =0; i < 7; i++)
    System.out.println(week[i]);
```

i=0		

Prints:

week

[0]	[1]	[2]	[3]	[4]	[5]	[6]
Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday

```
for (int i =0; i < 7; i++)
    System.out.println(week[i]);
```

i=0	0<7=true	

Prints:

week

[0]	[1]	[2]	[3]	[4]	[5]	[6]
Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday

```
for (int i =0; i < 7; i++)
    System.out.println(week[i]);
```

i=0	0<7=true	week[0]=Sunday

Prints:

week

[0]	[1]	[2]	[3]	[4]	[5]	[6]
Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday

```
for (int i =0; i < 7; i++)
    System.out.println(week[i]);
```

i=0	0<7=true	week[0]=Sunday

Prints:

Sunday

week

[0]	[1]	[2]	[3]	[4]	[5]	[6]
Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday

```
for (int i =0; i < 7; i++)
    System.out.println(week[i]);
```

i=0	0<7=true	week[0]=Sunday
i=1		

Prints:

Sunday

week

[0]	[1]	[2]	[3]	[4]	[5]	[6]
Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday

```
for (int i =0; i < 7; i++)
    System.out.println(week[i]);
```

i=0	0<7=true	week[0]=Sunday
i=1	1<7=true	

Prints:

Sunday

week

[0]	[1]	[2]	[3]	[4]	[5]	[6]
Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday

```
for (int i =0; i < 7; i++)
    System.out.println(week[i]);
```

i=0	0<7=true	week[0]=Sunday
i=1	1<7=true	week[1]=Monday

Prints:

Sunday

week

[0]	[1]	[2]	[3]	[4]	[5]	[6]
Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday

```
for (int i =0; i < 7; i++)
    System.out.println(week[i]);
```

i=0	0<7=true	week[0]=Sunday
i=1	1<7=true	week[1]=Monday

Prints:

Sunday

Monday

week

[0]	[1]	[2]	[3]	[4]	[5]	[6]
Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday

```
for (int i =0; i < 7; i++)
    System.out.println(week[i]);
```

i=0	0<7=true	week[0]=Sunday
i=1	1<7=true	week[1]=Monday
i=2		

Prints:

Sunday

Monday

week

[0]	[1]	[2]	[3]	[4]	[5]	[6]
Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday

```
for (int i =0; i < 7; i++)
    System.out.println(week[i]);
```

i=0	0<7=true	week[0]=Sunday
i=1	1<7=true	week[1]=Monday
i=2	2<7=true	

Prints:

Sunday

Monday

week

[0]	[1]	[2]	[3]	[4]	[5]	[6]
Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday

```
for (int i =0; i < 7; i++)
    System.out.println(week[i]);
```

i=0	0<7=true	week[0]=Sunday
i=1	1<7=true	week[1]=Monday
i=2	2<7=true	week[2]=Tuesday

Prints:

Sunday

Monday

Tuesday

week

[0]	[1]	[2]	[3]	[4]	[5]	[6]
Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday

```
for (int i =0; i < 7; i++)
    System.out.println(week[i]);
```

i=0	0<7=true	week[0]=Sunday
i=1	1<7=true	week[1]=Monday
i=2	2<7=true	week[2]=Tuesday
i=3		

Prints:

Sunday

Monday

Tuesday

week

[0]	[1]	[2]	[3]	[4]	[5]	[6]
Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday

```
for (int i =0; i < 7; i++)
    System.out.println(week[i]);
```

i=0	0<7=true	week[0]=Sunday
i=1	1<7=true	week[1]=Monday
i=2	2<7=true	week[2]=Tuesday
i=3	3<7=true	

Prints:

Sunday

Monday

Tuesday

week

[0]	[1]	[2]	[3]	[4]	[5]	[6]
Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday

```
for (int i =0; i < 7; i++)
    System.out.println(week[i]);
```

i=0	0<7=true	week[0]=Sunday
i=1	1<7=true	week[1]=Monday
i=2	2<7=true	week[2]=Tuesday
i=3	3<7=true	week[3]=Wednesday

Prints:

Sunday

Monday

Tuesday

Wednesday

week

[0]	[1]	[2]	[3]	[4]	[5]	[6]
Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday

```
for (int i =0; i < 7; i++)
    System.out.println(week[i]);
```

i=0	0<7=true	week[0]=Sunday
i=1	1<7=true	week[1]=Monday
i=2	2<7=true	week[2]=Tuesday
i=3	3<7=true	week[3]=Wednesday
i=4		

Prints:

Sunday

Monday

Tuesday

Wednesday

week

[0]	[1]	[2]	[3]	[4]	[5]	[6]
Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday

```
for (int i =0; i < 7; i++)
    System.out.println(week[i]);
```

i=0	0<7=true	week[0]=Sunday
i=1	1<7=true	week[1]=Monday
i=2	2<7=true	week[2]=Tuesday
i=3	3<7=true	week[3]=Wednesday
i=4	4<7=true	

Prints:

Sunday

Monday

Tuesday

Wednesday

week

[0]	[1]	[2]	[3]	[4]	[5]	[6]
Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday

```
for (int i =0; i < 7; i++)
    System.out.println(week[i]);
```

i=0	0<7=true	week[0]=Sunday
i=1	1<7=true	week[1]=Monday
i=2	2<7=true	week[2]=Tuesday
i=3	3<7=true	week[3]=Wednesday
i=4	4<7=true	week[4]=Thursday

Prints:

Sunday

Monday

Tuesday

Wednesday

Thursday

week

[0]	[1]	[2]	[3]	[4]	[5]	[6]
Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday

```
for (int i =0; i < 7; i++)
    System.out.println(week[i]);
```

i=0	0<7=true	week[0]=Sunday
i=1	1<7=true	week[1]=Monday
i=2	2<7=true	week[2]=Tuesday
i=3	3<7=true	week[3]=Wednesday
i=4	4<7=true	week[4]=Thursday
i=5		

Prints:

Sunday

Monday

Tuesday

Wednesday

Thursday

week

[0]	[1]	[2]	[3]	[4]	[5]	[6]
Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday

```
for (int i =0; i < 7; i++)
    System.out.println(week[i]);
```

i=0	0<7=true	week[0]=Sunday
i=1	1<7=true	week[1]=Monday
i=2	2<7=true	week[2]=Tuesday
i=3	3<7=true	week[3]=Wednesday
i=4	4<7=true	week[4]=Thursday
i=5	5<7=true	

Prints:

Sunday

Monday

Tuesday

Wednesday

Thursday

week

[0]	[1]	[2]	[3]	[4]	[5]	[6]
Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday

```
for (int i =0; i < 7; i++)
    System.out.println(week[i]);
```

i=0	0<7=true	week[0]=Sunday
i=1	1<7=true	week[1]=Monday
i=2	2<7=true	week[2]=Tuesday
i=3	3<7=true	week[3]=Wednesday
i=4	4<7=true	week[4]=Thursday
i=5	5<7=true	week[5]=Friday

Prints:

Sunday

Monday

Tuesday

Wednesday

Thursday

Friday

week

[0]	[1]	[2]	[3]	[4]	[5]	[6]
Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday

```
for (int i =0; i < 7; i++)
    System.out.println(week[i]);
```

i=0	0<7=true	week[0]=Sunday
i=1	1<7=true	week[1]=Monday
i=2	2<7=true	week[2]=Tuesday
i=3	3<7=true	week[3]=Wednesday
i=4	4<7=true	week[4]=Thursday
i=5	5<7=true	week[5]=Friday
i=6		

Prints:

Sunday

Monday

Tuesday

Wednesday

Thursday

Friday

week

[0]	[1]	[2]	[3]	[4]	[5]	[6]
Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday

```
for (int i =0; i < 7; i++)
    System.out.println(week[i]);
```

i=0	0<7=true	week[0]=Sunday
i=1	1<7=true	week[1]=Monday
i=2	2<7=true	week[2]=Tuesday
i=3	3<7=true	week[3]=Wednesday
i=4	4<7=true	week[4]=Thursday
i=5	5<7=true	week[5]=Friday
i=6	6<7=true	

Prints:

Sunday

Monday

Tuesday

Wednesday

Thursday

Friday

week

[0]	[1]	[2]	[3]	[4]	[5]	[6]
Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday

```
for (int i =0; i < 7; i++)
    System.out.println(week[i]);
```

i=0	0<7=true	week[0]=Sunday
i=1	1<7=true	week[1]=Monday
i=2	2<7=true	week[2]=Tuesday
i=3	3<7=true	week[3]=Wednesday
i=4	4<7=true	week[4]=Thursday
i=5	5<7=true	week[5]=Friday
i=6	6<7=true	week[6]=Saturday

Prints:

Sunday

Monday

Tuesday

Wednesday

Thursday

Friday

Saturday

week

[0]	[1]	[2]	[3]	[4]	[5]	[6]
Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday

```
for (int i =0; i < 7; i++)
    System.out.println(week[i]);
```

i=0	0<7=true	week[0]=Sunday
i=1	1<7=true	week[1]=Monday
i=2	2<7=true	week[2]=Tuesday
i=3	3<7=true	week[3]=Wednesday
i=4	4<7=true	week[4]=Thursday
i=5	5<7=true	week[5]=Friday
i=6	6<7=true	week[6]=Saturday
i=7		

Prints:

Sunday

Monday

Tuesday

Wednesday

Thursday

Friday

Saturday

week

[0]	[1]	[2]	[3]	[4]	[5]	[6]
Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday

```
for (int i =0; i < 7; i++)
    System.out.println(week[i]);
```

i=0	0<7=true	week[0]=Sunday
i=1	1<7=true	week[1]=Monday
i=2	2<7=true	week[2]=Tuesday
i=3	3<7=true	week[3]=Wednesday
i=4	4<7=true	week[4]=Thursday
i=5	5<7=true	week[5]=Friday
i=6	6<7=true	week[6]=Saturday
i=7	7<7=false	

Prints:

Sunday

Monday

Tuesday

Wednesday

Thursday

Friday

Saturday


The End
is Near
DAN CHUSID 2003

Array Characteristics

- Allow you to use **loops** so they save a lot of lines of code.
- MUCH more **efficient** than single variables for big sets of data.
- Are **organized**, one thing per space.
- All elements in the array have the **same type**.
- Have a **fixed size** from the beginning.

Declaring:

```
String hScores [ ] = new String [3];

int checkers [ ][ ] = new int [8] [8];

String week[ ] = {“Sunday”, “Monday”, “Tuesday”, “Wednesday”, “Thursday”, “Friday”, “Saturday”};
```

Printing:

```
for (int i=0; i<week.length; i++)
    System.out.println(week[i]);
```