

Wednesday February 13

Lecture 12

Computational Problem: Printing a Comma-Separated List

```

System.out.print("Names:")
for(int i = 0; i < names.length; i++) {
    System.out.print(names[i]);
    if (i < names.length - 1)
        System.out.print(", ");
}
System.out.println("");
    
```

Alan, Mark, Tom

print vs. println

Console
Names: Alan, Mark, Tom.

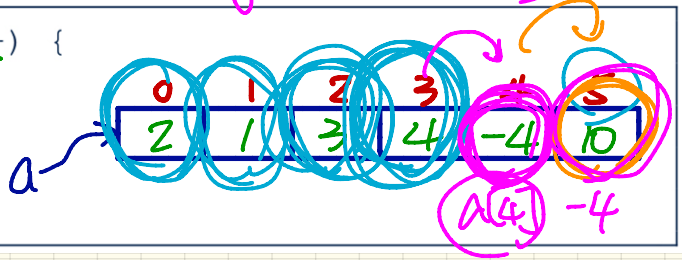
i	i < names.length	names[i]	i < names.length - 1
0	0 < 3 T	"Alan"	0 < 2 T
1	1 < 3 T	"Mark"	1 < 2 T
2	2 < 3 T	"Tom"	2 < 2 (F)
3	3 < 3 (F)		

Computational Problem: Conditional Printing

$i < a.length$ && $a[i] > 0$

6

```
for(int i = 0; i < a.length; i++) {  
    if (a[i] > 0) {  
        System.out.println(a[i]);  
    }  
    if (i < a.length - 1) {  
        print(" ");  
    }  
}
```



Console

```
2  
1  
3  
4  
10
```

i	$i < a.length$	$a[i]$	$a[i] > 0$
0	$0 < 6$ T	2	$2 > 0$ T
1		1	$1 > 0$ T
2		3	$3 > 0$ T
3		4	$4 > 0$ T
4	$4 < 6$ T	-4	$-4 > 0$ F
5	$5 < 6$	10	$10 > 0$ T

Computational Problem: Finding Maximum

$$\text{max} = a[z]$$

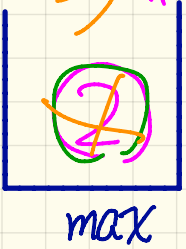
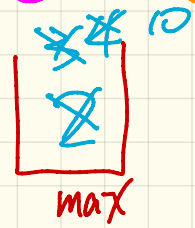
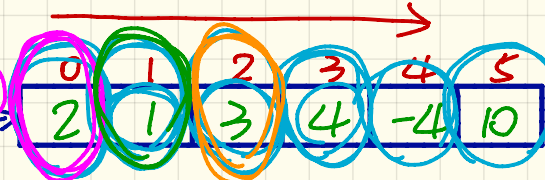
```

1 → int max = a[0];
2 for (int i = 0; i < a.length; i++) {
3   if (a[i] > max) { max = a[i]; }
4 }
5 System.out.println("Maximum is " + max);
    
```

$a[0] > a[0] \rightarrow F$
- always

the very 1st iteration always bypasses the body of if statement

i	i < a.length	a[i]
0	0 < 6 T	a[0] > 2 F
1	1 < 6 T	a[1] > 2 F
2	2 < 6 T	a[2] > 2 T



Computational Problem: Finding Maximum

Q: What if we change the initialization in **L1** to `int max = 0`?

```

1 int max = 0;
2 for(int i = 0; i < a.length; i++) {
3     if (a[i] > max) { max = a[i]; }
4 }
5 System.out.println("Maximum is " + max);
    
```

Annotations:
 - Red box around `a.length` with '3' above it.
 - Pink box around `a[i] > max`.
 - Red box around `max = a[i];`.
 - Blue circles around '0' in `max` and '0' in `max` in the print statement.
 - Blue arrow pointing to `max` in the print statement with 'logical error' written next to it.
 - Blue 'X' marks over `max` in the print statement.

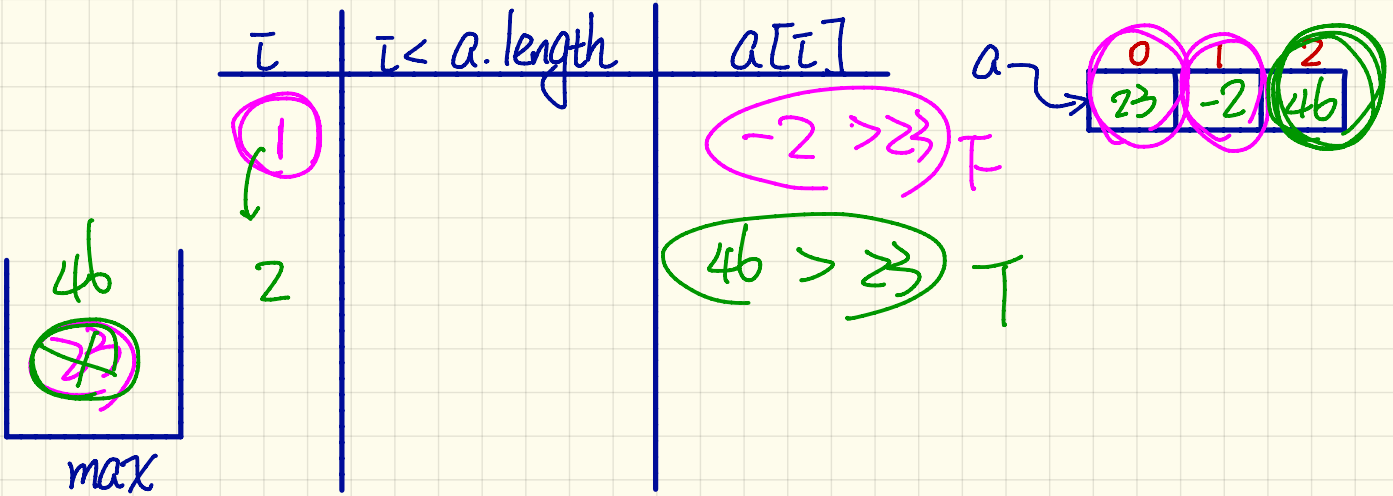
i	i < a.length	a[i]	a[i] > max	max
0	0 < 3 T	-3	-3 > 0 F	0
1	1 < 3 T	-4	-4 > 0 F	0
2	2 < 3 T	-1	-1 > 0 F	0

Additional visualizations:
 - A box containing the array [0, -4, -1] with indices 0, 1, 2 above each element.
 - A box containing the value 0, labeled 'max' below it.
 - Blue circles around the '0' in the 'max' box and the '0' in the array box.
 - Pink circles around the '-4' and '-1' in the array box.
 - Red circles around the '-3' in the array box.

Computational Problem: Finding Maximum

Q: What if we change the initialization in **L2** to `int i = 1`?

```
1 int max = a[0]; // position 0 is already handled
2 for(int i = 1; i < a.length; i++) {
3     if (a[i] > max) { max = a[i]; } // max = a[2]
4 }
5 System.out.println("Maximum is " + max); // 46
```



```

1 int max = a[0];
2 for(int i = 0; i < a.length; i++) {
3     if (a[i] > max) { max = a[i]; }
4 }
5 System.out.println("Maximum is " + max);

```

$a.length - 1$

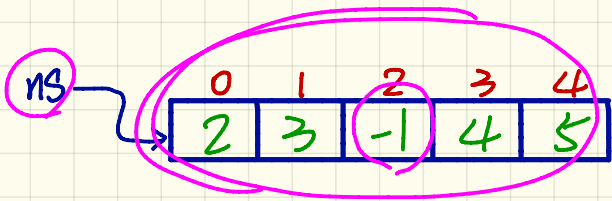
$i + 1$

logically
incorrect
even though
it samples

last iteration

$i: a.length - 1$
 $a[(a.length - 1) + 1]$

False



answer

```
boolean allPositive =  
    &&  
    &&  
    &&  
    &&  
    ns[0] > 0  
    ns[1] > 0  
    ns[2] > 0  
    ns[3] > 0  
    ns[4] > 0
```


Computational Problem: Are all numbers positive?

```

1 int[] ns = {2, 3, -1, 4, 5};
2 boolean soFarOnlyPosNums = true;
3 int i = 0;
4 while (i < ns.length) {
5     soFarOnlyPosNums = soFarOnlyPosNums && ns[i] > 0;
6     i = i + 1;
7 }
    
```

Annotations: **Version 1** (green box), **stopn = stopn && ns[i] > 0** (red circles), **stopn = stopn && (ns[0] > 0)** (blue circles), **stopn = stopn && (ns[3] > 0)** (pink circles).

i	$i < ns.length$	$ns[i] > 0$	
0	$0 < 5$ T	T	F
1	$1 < 5$ T	F	F
2	$2 < 5$ T	F	F
3	$3 < 5$ T	T	F

Diagram: **ns** array [2, 3, -1, 4, 5] with indices 0-4. **soFarOnlyPosNums** is shown as **-1** (F) at the bottom. Annotations include **stopn = stopn && ns[i] > 0** and **stopn = stopn && (ns[3] > 0)** with various truth value annotations (true/false).