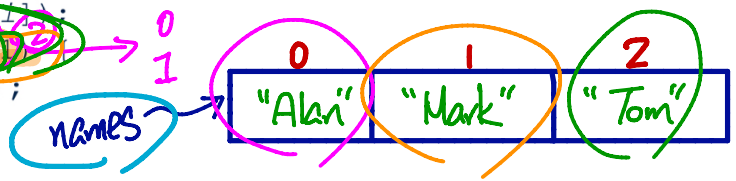
Wednesday February 13

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.

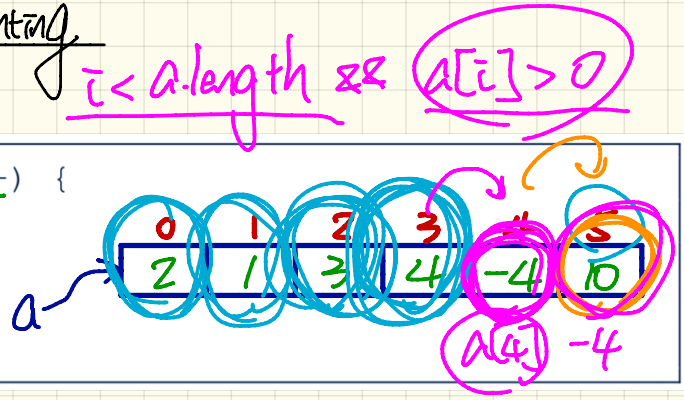
i	i < names.length		names[i]	i < names.length - 1	
0	0 < 3	T	"Alan"	0 < 2	T
1	1 < 3	T	"Mark"	1 < 2	T
2	2 < 3	T	"Tom"	2 < 2	F
3	3 < 3	F			

Computational Problem : Conditional Printing

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

b

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




```
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

max = a[i]

```
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);
```

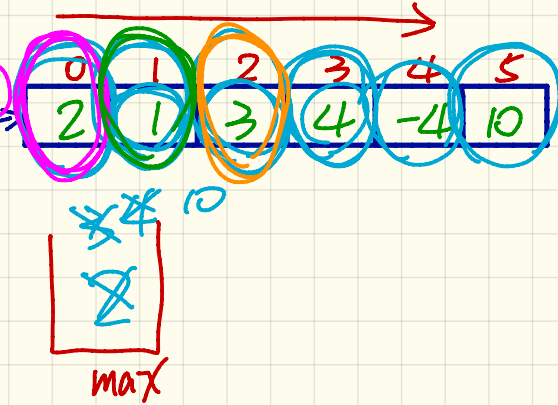
a[0] > a[0] → F - always

b

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

the very 1st iteration always by passes the body of statement

max



Computational Problem: Finding Maximum

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

```
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);
```

0 3 ✗ ✗ ✗ 0 ✗ logical error

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

a → [0: -3] [1: -4] [2: -1]

0 → [0] max

Computational Problem : Finding Maximum

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

```
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);
```

position 0 is already handled

max = a[2]

46

i	i < a.length	a[i]
1		-2 > 23 F
2		46 > 23 T

a → [23, -2, 46] (0, 1, 2)

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 though even it compiles

i

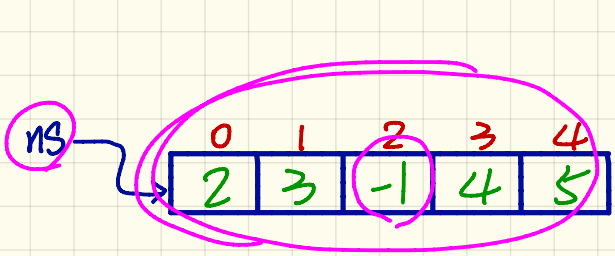
last iteration

i: a.length - 1

a[(a.length - 1) + 1]

a.length

i



answer

i

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  }
```

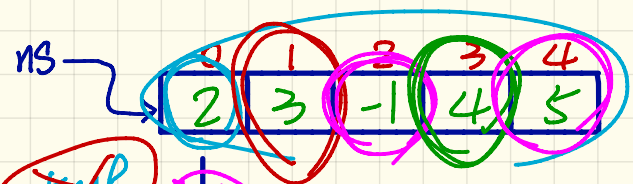
Version 1

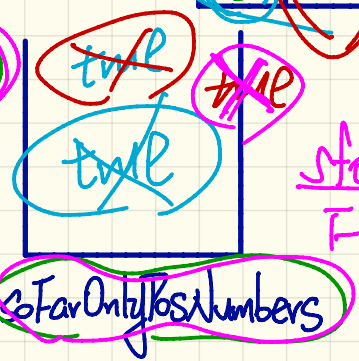
soFarOnlyPosNums = soFarOnlyPosNums && (ns[i] > 0);

soFarOnlyPosNums = soFarOnlyPosNums && (ns[0] > 0);

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

soFarOnlyPosNums = soFarOnlyPosNums && a[3] > 0;





soFarOnlyPosNums = soFarOnlyPosNums && a[2] > 0
-1
F