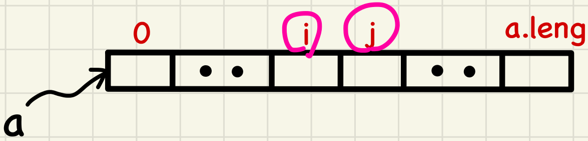
Sorting Orders of Arrays



decreasing $a[i] > a[j]$

ascending $a[i] < a[j]$

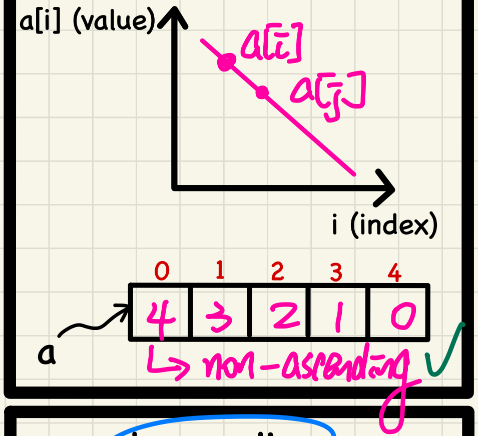
non-descending $!(a[i] > a[j])$

$\equiv a[i] \leq a[j]$

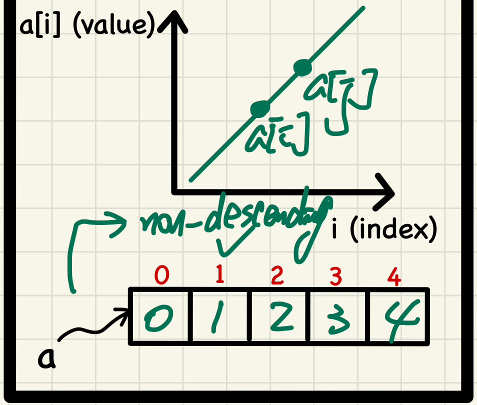
non-ascending $!(a[i] < a[j])$

$\equiv a[i] \geq a[j]$

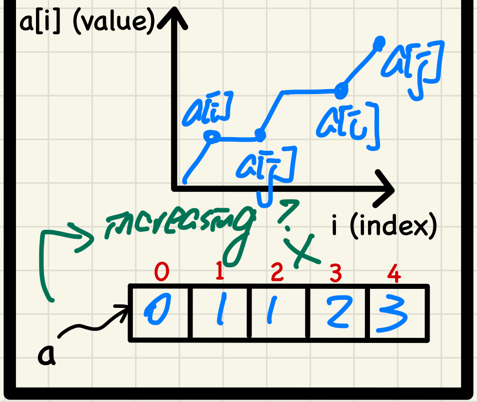
decreasing/descending



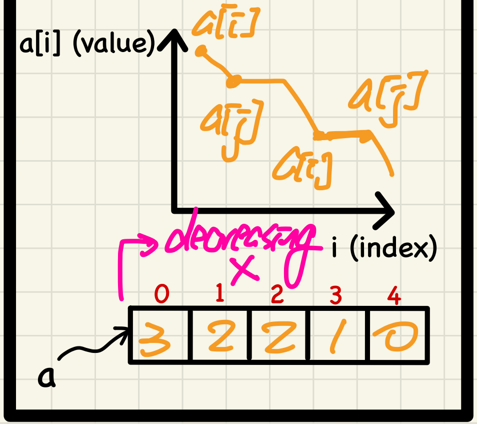
increasing/ascending



non-descending



non-ascending



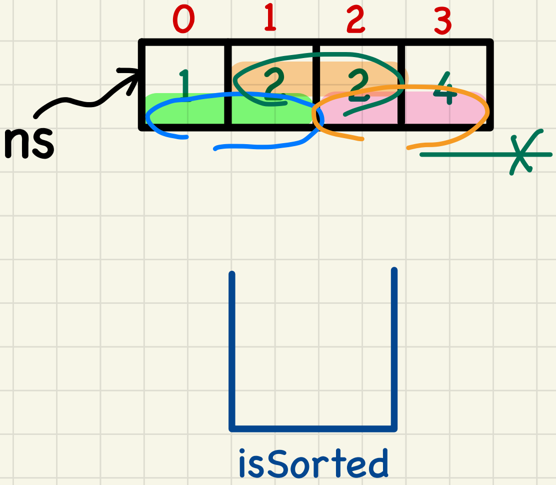
Computational Problem: Is an Array Sorted?

3

```
boolean isSorted = true;
for(int i = 0; isSorted && i < a.length - 1; i ++) {
  isSorted = a[i] <= a[i + 1];
}
```

Test Case 1

④



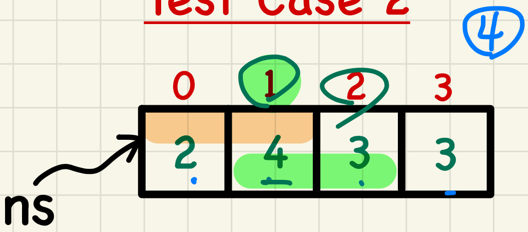
i	i < a.length	a[i] <= a[i + 1]
0		$1 \le 2$ T
1		$2 \le 2$ T
2		$2 \le 4$ T
3	T	

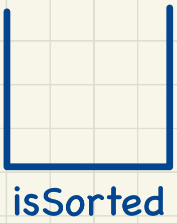
Computational Problem: Is an Array Sorted?

```
boolean isSorted = true;
for(int i = 0; isSorted && i < a.length - 1; i ++) {
  isSorted = a[i] <= a[i + 1];
}
```

F F F

Test Case 2





i	i < a.length	a[i] <= a[i + 1]
0		2 ≤ 4 T
1		4 ≤ 3 F
~~2~~		

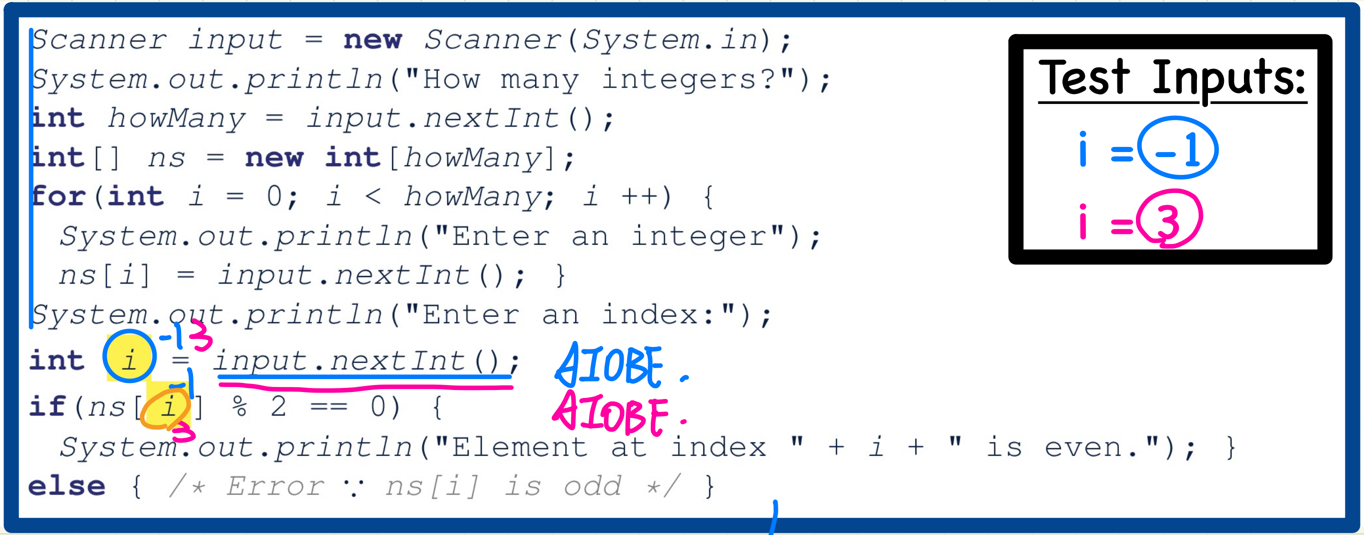
exit

Lecture 3

Part F

Loops and Arrays -
Short Circuit Evaluation and Indexing

Unguarded Array Indexing



```
Scanner input = new Scanner(System.in);
System.out.println("How many integers?");
int howMany = input.nextInt();
int[] ns = new int[howMany];
for(int i = 0; i < howMany; i ++) {
  System.out.println("Enter an integer");
  ns[i] = input.nextInt(); }
System.out.println("Enter an index:");
int i = input.nextInt();
if(ns[i] % 2 == 0) {
  System.out.println("Element at index " + i + " is even."); }
else { /* Error ∵ ns[i] is odd */ }
```

-1 3

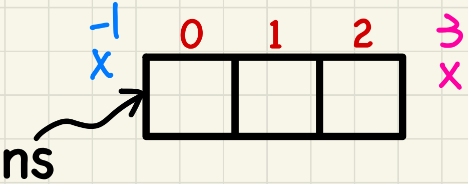
AIOBE.

AIOBE.

Test Inputs:

i = -1

i = 3



resolution to AIOBE

SCE.

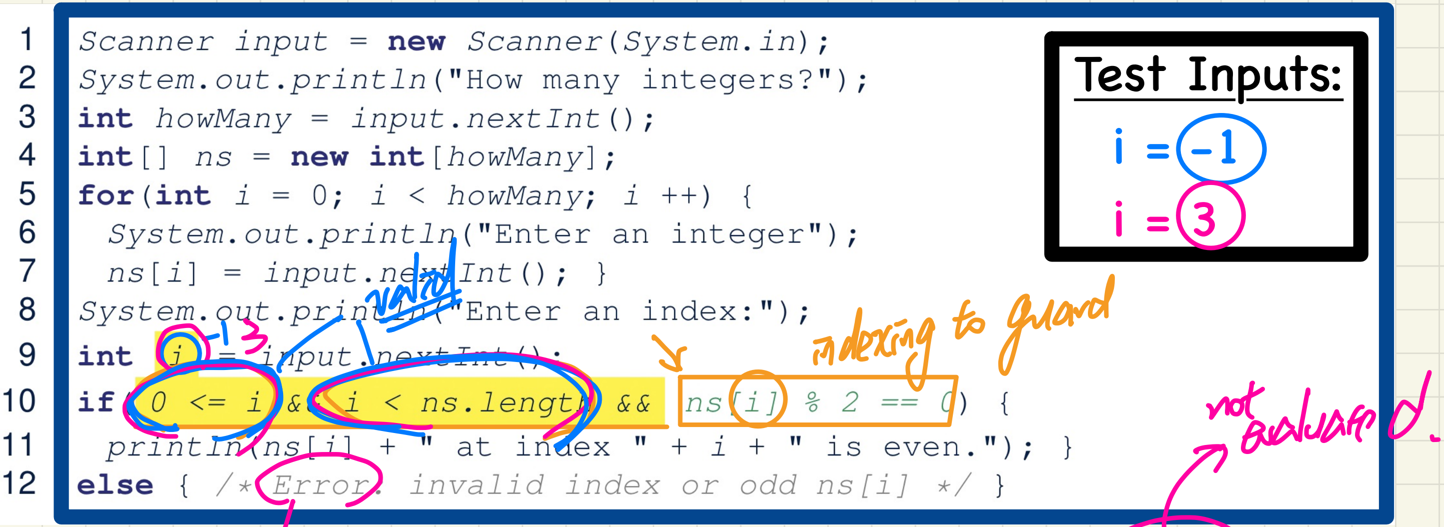
&& } guard array indexing

||

ns[i]

Use of Conjunction (&&)

Guarding Array Indexing using Short Circuit



```
Scanner input = new Scanner(System.in);
System.out.println("How many integers?");
int howMany = input.nextInt();
int[] ns = new int[howMany];
for(int i = 0; i < howMany; i ++) {
  System.out.println("Enter an integer");
  ns[i] = input.nextInt(); }
System.out.println("Enter an index:");
int i = input.nextInt();
if(0 <= i && i < ns.length && ns[i] % 2 == 0) {
  println(ns[i] + " at index " + i + " is even."); }
else { /* Error: invalid index or odd ns[i] */ }
```

Test Inputs:

i = -1

i = 3

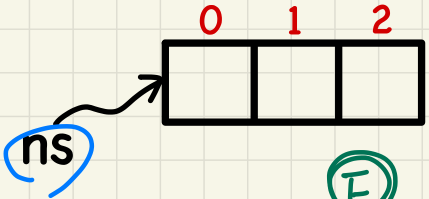
-1 3

valid

indexing to guard

not evaluated.

Exception.



$\underline{0 <= 3}$ && $\underline{3 < 3}$ $\boxed{\text{\&\& } ns[3] \% == 0.}$

T F

bypassed

will not be evaluated.

E

bypassed.

$\underline{0 <= -1}$ $\boxed{\text{\&\& } -1 < 3 \text{ \&\& } ns[-1] \% 2 == 0}$

Use of Disjunction (||)

Guarding Array Indexing using Short Circuit

```
Scanner input = new Scanner(System.in);
System.out.println("How many integers?");
int howMany = input.nextInt();
int[] ns = new int[howMany];
for(int i = 0; i < howMany; i ++) {
  System.out.println("Enter an integer");
  ns[i] = input.nextInt(); }
System.out.println("Enter an index:");
int i = input.nextInt();
if( i < 0 || i >= ns.length || ns[i] % 2 == 1) {
  /* Error: invalid index or odd ns[i] */ }
else { println(ns[i] + " at index " + i + " is even."); }
```

invalid

③

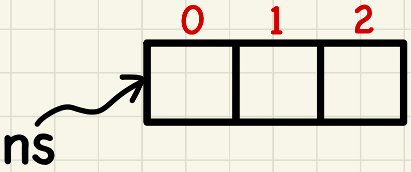
to guard.

bypassed

Test Inputs:

i = -1

i = 3



$\underline{3 < 0} \;||\; \underline{3 \geq 3} \;||\; \boxed{ns[3] \% 2 == 1}$

F

T

not evaluated.

bypassed

$\underline{-1 < 0} \;||\; \boxed{-1 \geq 3 \;||\; ns[-1] \% 2 == 1}$

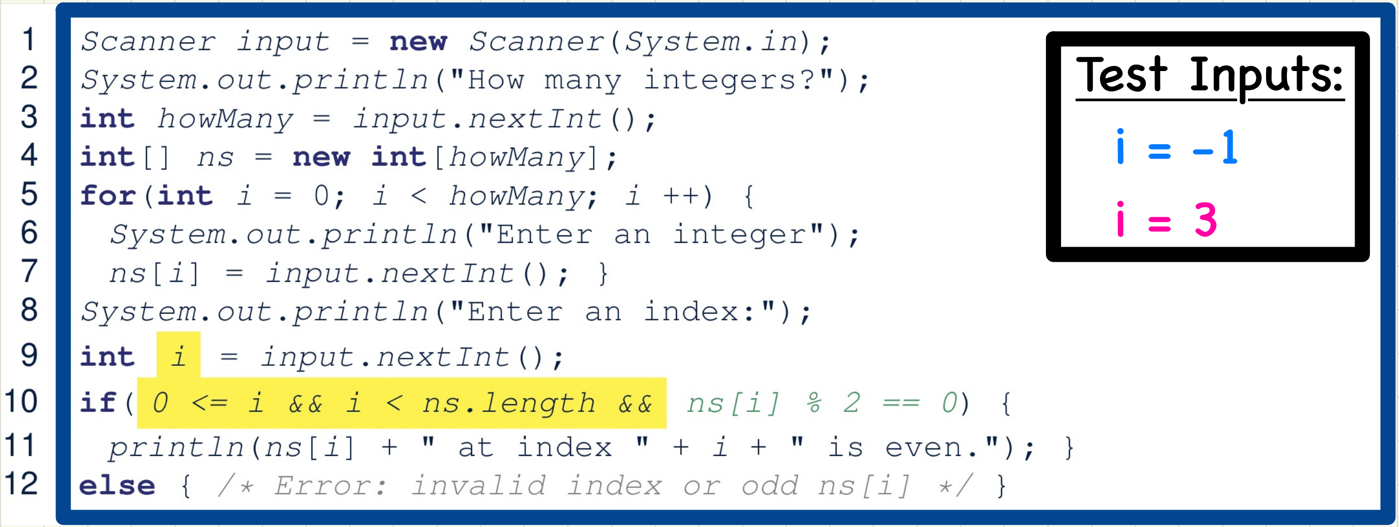
T

invalid

bypassed

not evaluated.

Guarding Array Indexing using Short Circuit



```
1  Scanner input = new Scanner(System.in);
2  System.out.println("How many integers?");
3  int howMany = input.nextInt();
4  int[] ns = new int[howMany];
5  for(int i = 0; i < howMany; i ++) {
6    System.out.println("Enter an integer");
7    ns[i] = input.nextInt(); }
8  System.out.println("Enter an index:");
9  int i = input.nextInt();
10 if( 0 <= i && i < ns.length && ns[i] % 2 == 0) {
11   println(ns[i] + " at index " + i + " is even."); }
12 else { /* Error: invalid index or odd ns[i] */ }
```

Test Inputs:

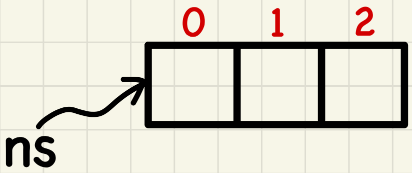
i = -1

i = 3

① work

② crash?

Q. L10: 0 <= i && ns[i] % 2 == 0 && i < ns.length?



Use of Conjunction (&&)

Exercise

Guarding Array Indexing using Short Circuit

```
1  Scanner input = new Scanner(System.in);
2  System.out.println("How many integers?");
3  int howMany = input.nextInt();
4  int[] ns = new int[howMany];
5  for(int i = 0; i < howMany; i ++) {
6    System.out.println("Enter an integer");
7    ns[i] = input.nextInt(); }
8  System.out.println("Enter an index:");
9  int i = input.nextInt();
10 if( i < 0 || i >= ns.length || ns[i] % 2 == 1) {
11   /* Error: invalid index or odd ns[i] */ }
12 else { println(ns[i] + " at index " + i + " is even."); }
```

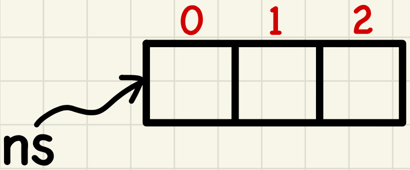
Test Inputs:

i = -1

i = 3

① crash ?
② work ?

Q. L10: i < 0 || ns[i] % 2 == 0 || i >= ns.length?



Use of Disjunction (||)

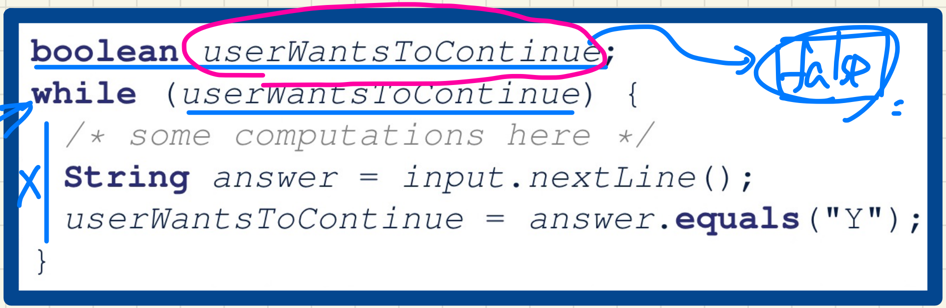
Exercise

Lecture 3

Part G

Loops and Arrays - Common Errors

Common Errors: Improper Initialization of Loop Counter



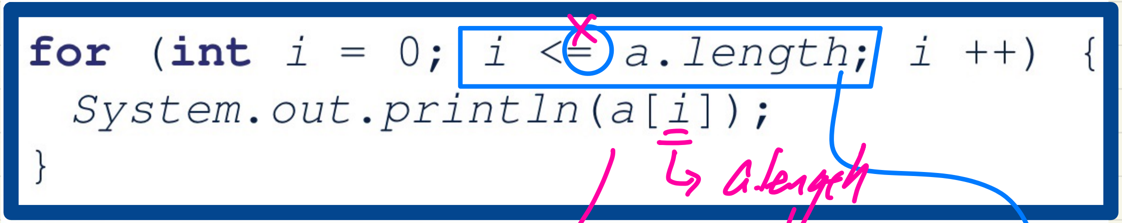
```
boolean userWantsToContinue;
while (userWantsToContinue) {
  /* some computations here */
  String answer = input.nextLine();
  userWantsToContinue = answer.equals("Y");
}
```

false

nothing will be executed

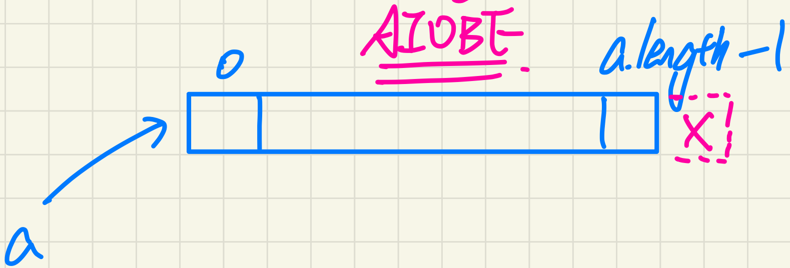
fix: boolean userWantsToContinue = true;

Common Errors: Improper Stay Condition



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

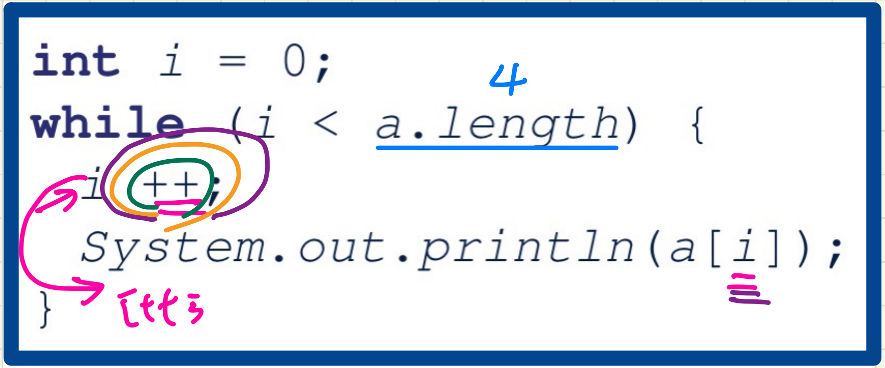
i ↳ a.length



i == a.length.

↳ iteration.

Common Errors: Improper Update to Loop Counter

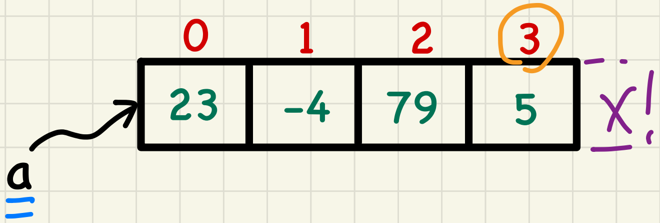


```
int i = 0;
while (i < a.length) {
  i++;
  System.out.println(a[i]);
}
```

4

fix

i++;

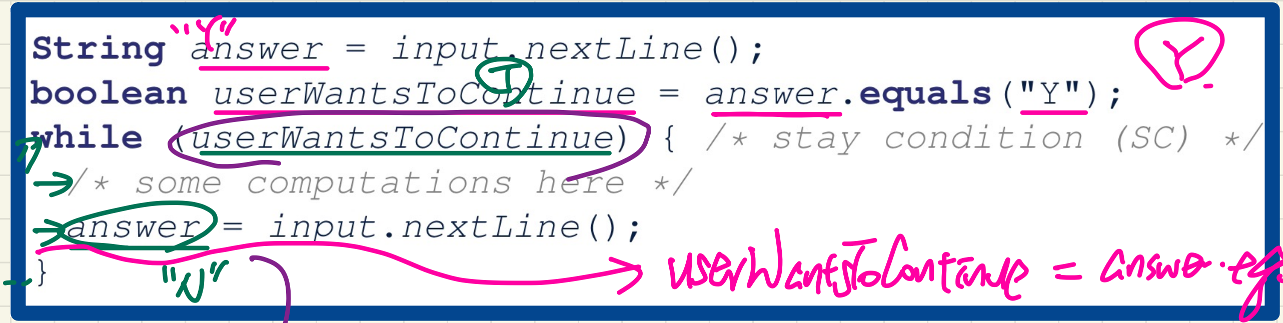


i	a[i]
0	a[1]
1	a[2]
2	a[3]
3	a[4]

AIOBE

X

Common Errors: Improper Update to Stay Condition



```java
String answer = input.nextLine();
boolean userWantsToContinue = answer.equals("Y");
while (userWantsToContinue) { /* stay condition (SC) */
  /* some computations here */
  answer = input.nextLine();
}
```

"Y"

T

Y

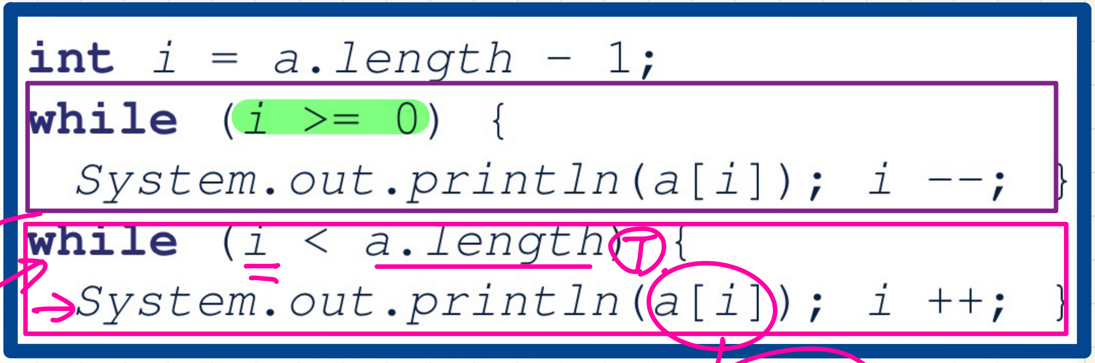
"N"

userWantsToContinue = answer.equals("Y");

Logical Error: infinite loop if 1st iteration allowed

∵ userWantsToContinue not updated.

Common Errors: Improper Initial Value of Loop Counter

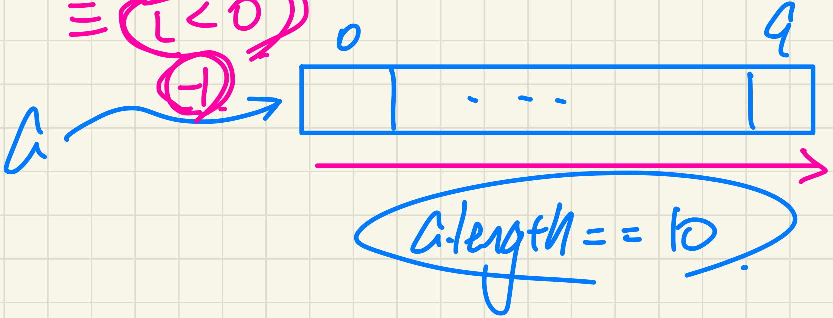


```
int i = a.length - 1;
while (i >= 0) {
  System.out.println(a[i]); i --; }
while (i < a.length) {
  System.out.println(a[i]); i ++; }
```

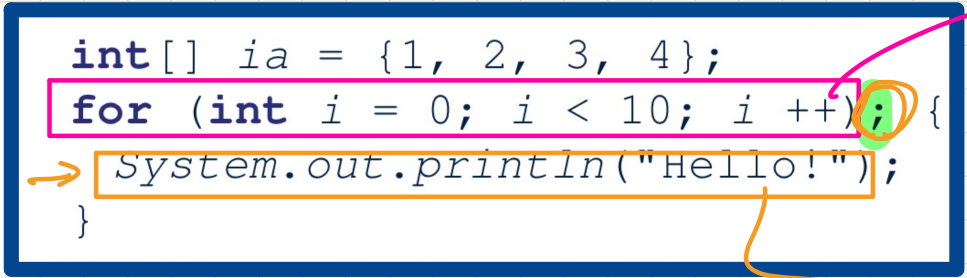
a[-1] ⇒ AIOBE

exit: !(i >= 0)

≡ i < 0



Common Errors: Misplaced Semicolon



```
int[] ia = {1, 2, 3, 4};
for (int i = 0; i < 10; i ++); {
  System.out.println("Hello!");
}
```

entire loop

not body of the loop.

Console

```
Hello
```