

Introduction to Java

Discussion 02

Announcements

1. Lab 1, Lab 2, and HW 0 due Friday 01/28 (all of these CANNOT be dropped)
2. HW 1 released Tuesday at noon, due next Tuesday 02/01
3. OH starts this week entirely online
4. Please complete the Pre-Semester Survey!

All About Your TA!

Edit this slide to include an intro about yourself.

Review

Anatomy of a Function

```
/** Print all primes up to and including LIMIT. */  
private static void printPrimes(int limit) {  
    for (int p = 2; p <= limit; p += 1) {  
        if (isPrime(p)) {  
            System.out.print(p + " ");  
        }  
    }  
    System.out.println();  
}
```

Comments

Keywords for the basic elements of the language (we will cover more later)

Type declarations - Java is statically typed so we have to tell the computer what type of value every variable holds and what every function returns

Variable and Function Names that allow us to refer to our stored values

Don't forget the brackets and semicolons!

Structure of a Class

```
public class CS61BStudent { // Class Declaration
    public int idNumber; // Instance Variables
    public int grade;
    public static String professor = "Hilfinger"; // Class (Static) Variables
    public CS61BStudent (int id) { // Constructor
        this.idNumber = id;
        this.grade = 100;
    }
    public void watchLecture() { // Instance Method
        ...
    }
    public static void updateGrades() { // Class (Static) Method
        ...
    }
}
```

Instantiating Classes

```
public class CS61BLauncher {  
    public static void main(String[] args) {  
        CS61BStudent studentOne; // Declare class  
        studentOne = new CS61BStudent(32259); // Instantiate and assign class  
        CS61BStudent studentTwo = new CS61BStudent(19234); // Both at once  
  
        studentOne.watchLecture(); // Instance methods are called on instance  
  
        CS61BStudent.updateGrades(); // Static methods can be  
                                     // called by class OR instance  
    }  
}
```

Static vs. Instance

Static variables and functions belong to the whole class.

Example: Every 61B Student shares the same professor, and if the professor were to change it would change for everyone.

Instance variables and functions belong to each individual instance.

Example: Each 61B Student has their own ID number, and changing a student's ID number doesn't change anything for any other student.

Worksheet

1 Old Town Code

```
1  int x = 7;
2  String chorus = "Thank u, next";
3  Singer queen = new Singer("Ariana");
4
5  while (x > 0) {
6      x -= 1;
7      queen.sing(chorus);
8  }
9
10 String[] phrases = {"love", "patience", "pain", "what does the fox say?"};
11
12 for (int i = 0; i < 3; i += 1) {
13     System.out.println("One taught me " + phrases[i]);
14 }
15
16 System.out.println(phrases[phrases.length - 1]);
```

What does this output?

1 Old Town Code

```
1  int x = 7; // Declares var x and assigns 7 to it
2  String chorus = "Thank u, next";
3  Singer queen = new Singer("Ariana");
4
5  while (x > 0) {
6      x -= 1;
7      queen.sing(chorus);
8  }
9
10 String[] phrases = {"love", "patience", "pain", "what does the fox say?"};
11
12 for (int i = 0; i < 3; i += 1) {
13     System.out.println("One taught me " + phrases[i]);
14 }
15
16 System.out.println(phrases[phrases.length - 1]);
```

What does this output?

1 Old Town Code

```
1  int x = 7; // Declares var x and assigns 7 to it
2  String chorus = "Thank u, next"; // Declares var chorus and assigns a String to it
3  Singer queen = new Singer("Ariana");
4
5  while (x > 0) {
6      x -= 1;
7      queen.sing(chorus);
8  }
9
10 String[] phrases = {"love", "patience", "pain", "what does the fox say?"};
11
12 for (int i = 0; i < 3; i += 1) {
13     System.out.println("One taught me " + phrases[i]);
14 }
15
16 System.out.println(phrases[phrases.length - 1]);
```

What does this output?

1 Old Town Code

```
1  int x = 7; // Declares var x and assigns 7 to it
2  String chorus = "Thank u, next"; // Declares var chorus and assigns a String to it
3  Singer queen = new Singer("Ariana"); // Declares var queen and assigns a Singer to it
4
5  while (x > 0) {
6      x -= 1;
7      queen.sing(chorus);
8  }
9
10 String[] phrases = {"love", "patience", "pain", "what does the fox say?"};
11
12 for (int i = 0; i < 3; i += 1) {
13     System.out.println("One taught me " + phrases[i]);
14 }
15
16 System.out.println(phrases[phrases.length - 1]);
```

What does this output?

1 Old Town Code

```
1  int x = 7; // Declares var x and assigns 7 to it
2  String chorus = "Thank u, next"; // Declares var chorus and assigns a String to it
3  Singer queen = new Singer("Ariana"); // Declares var queen and assigns a Singer to it
4
5  while (x > 0) { // Checks if x is still greater than 0
6      x -= 1; // If so it deducts 1 from x
7      queen.sing(chorus); // And it calls the queen.sing method on chorus
8  }
9
10 String[] phrases = {"love", "patience", "pain", "what does the fox say?"};
11
12 for (int i = 0; i < 3; i += 1) {
13     System.out.println("One taught me " + phrases[i]);
14 }
15
16 System.out.println(phrases[phrases.length - 1]);
```

What does this output?

1 Old Town Code

```
1  int x = 7; // Declares var x and assigns 7 to it
2  String chorus = "Thank u, next"; // Declares var chorus and assigns a String to it
3  Singer queen = new Singer("Ariana"); // Declares var queen and assigns a Singer to it
4
5  while (x > 0) { // Checks if x is still greater than 0
6      x -= 1; // If so it deducts 1 from x
7      queen.sing(chorus); // And it calls the queen.sing method on chorus
8  }
9
10 String[] phrases = {"love", "patience", "pain", "what does the fox say?"};
11 // Declares var phrases and assigns an array of Strings to it
12 for (int i = 0; i < 3; i += 1) {
13     System.out.println("One taught me " + phrases[i]);
14 }
15
16 System.out.println(phrases[phrases.length - 1]);
```

What does this output?

1 Old Town Code

```
1  int x = 7; // Declares var x and assigns 7 to it
2  String chorus = "Thank u, next"; // Declares var chorus and assigns a String to it
3  Singer queen = new Singer("Ariana"); // Declares var queen and assigns a Singer to it
4
5  while (x > 0) { // Checks if x is still greater than 0
6      x -= 1; // If so it deducts 1 from x
7      queen.sing(chorus); // And it calls the queen.sing method on chorus
8  }
9
10 String[] phrases = {"love", "patience", "pain", "what does the fox say?"};
11 // Declares var phrases and assigns an array of Strings to it
12 for (int i = 0; i < 3; i += 1) { // Declares i and checks if its still less than 3
13     System.out.println("One taught me " + phrases[i]); // If so it prints
14 } // And increments by one
15
16 System.out.println(phrases[phrases.length - 1]);
```

What does this output?

1 Old Town Code

```
1  int x = 7; // Declares var x and assigns 7 to it
2  String chorus = "Thank u, next"; // Declares var chorus and assigns a String to it
3  Singer queen = new Singer("Ariana"); // Declares var queen and assigns a Singer to it
4
5  while (x > 0) { // Checks if x is still greater than 0
6      x -= 1; // If so it deducts 1 from x
7      queen.sing(chorus); // And it calls the queen.sing method on chorus
8  }
9
10 String[] phrases = {"love", "patience", "pain", "what does the fox say?"};
11 // Declares var phrases and assigns an array of Strings to it
12 for (int i = 0; i < 3; i += 1) { // Declares i and checks if its still less than 3
13     System.out.println("One taught me " + phrases[i]); // If so it prints
14 } // And increments by one
15
16 System.out.println(phrases[phrases.length - 1]); // Prints the last phrase in phrases
```

What does this output?

1 Old Town Code

```
1  int x = 7;
2  String chorus = "Thank u, next";
3  Singer queen = new Singer("Ariana");
4
5  while (x > 0) {
6      x -= 1;
7      queen.sing(chorus);
8  }
9
10 String[] phrases = {"love", "patience", "pain", "what does the fox say?"};
11
12 for (int i = 0; i < 3; i += 1) {
13     System.out.println("One taught me " + phrases[i]);
14 }
15
16 System.out.println(phrases[phrases.length - 1]);
```

Console Output

```
One taught me love
One taught me patience
One taught me pain
What does the fox say?
```

What does this output?

2 Reading Code: A Mystery

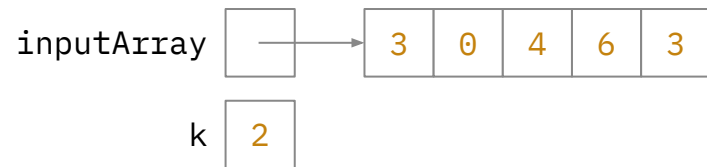
```
1 public static int mystery1(int[] inputArray, int k) {
2     int x = inputArray[k];
3     int answer = k;
4     int index = k + 1;
5     while (index < inputArray.length) {
6         if (inputArray[index] < x) {
7             x = inputArray[index];
8             answer = index;
9         }
10        index = index + 1;
11    }
12    return answer;
13 }
```

What does this return when the input array is [3, 0, 4, 6, 3] and k is 2?

What does this answer mean?

2 Reading Code: A Mystery

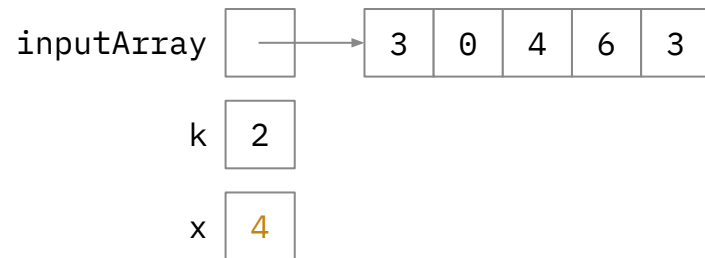
```
1 public static int mystery1(int[] inputArray, int k) {
2     int x = inputArray[k];
3     int answer = k;
4     int index = k + 1;
5     while (index < inputArray.length) {
6         if (inputArray[index] < x) {
7             x = inputArray[index];
8             answer = index;
9         }
10        index = index + 1;
11    }
12    return answer;
13 }
```



**What does this return when the input array is [3, 0, 4, 6, 3] and k is 2?
What does this answer mean?**

2 Reading Code: A Mystery

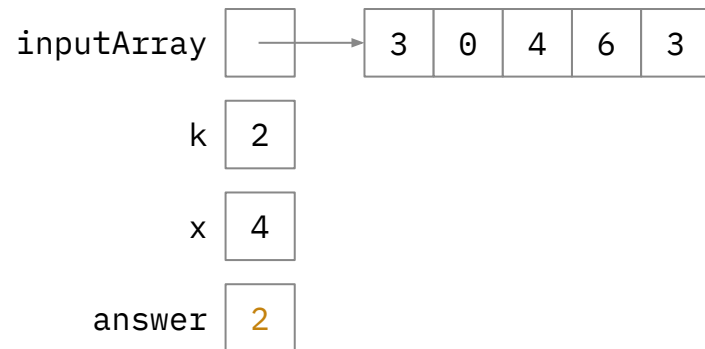
```
1 public static int mystery1(int[] inputArray, int k) {
2     int x = inputArray[k];
3     int answer = k;
4     int index = k + 1;
5     while (index < inputArray.length) {
6         if (inputArray[index] < x) {
7             x = inputArray[index];
8             answer = index;
9         }
10        index = index + 1;
11    }
12    return answer;
13 }
```



**What does this return when the input array is [3, 0, 4, 6, 3] and k is 2?
What does this answer mean?**

2 Reading Code: A Mystery

```
1 public static int mystery1(int[] inputArray, int k) {
2     int x = inputArray[k];
3     int answer = k;
4     int index = k + 1;
5     while (index < inputArray.length) {
6         if (inputArray[index] < x) {
7             x = inputArray[index];
8             answer = index;
9         }
10        index = index + 1;
11    }
12    return answer;
13 }
```



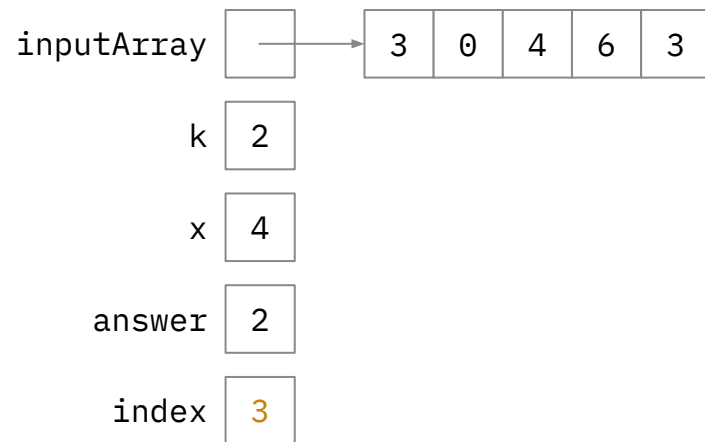
**What does this return when the input array is [3, 0, 4, 6, 3] and k is 2?
What does this answer mean?**

2 Reading Code: A Mystery

```

1  public static int mystery1(int[] inputArray, int k) {
2      int x = inputArray[k];
3      int answer = k;
4      int index = k + 1;
5      while (index < inputArray.length) {
6          if (inputArray[index] < x) {
7              x = inputArray[index];
8              answer = index;
9          }
10         index = index + 1;
11     }
12     return answer;
13 }

```



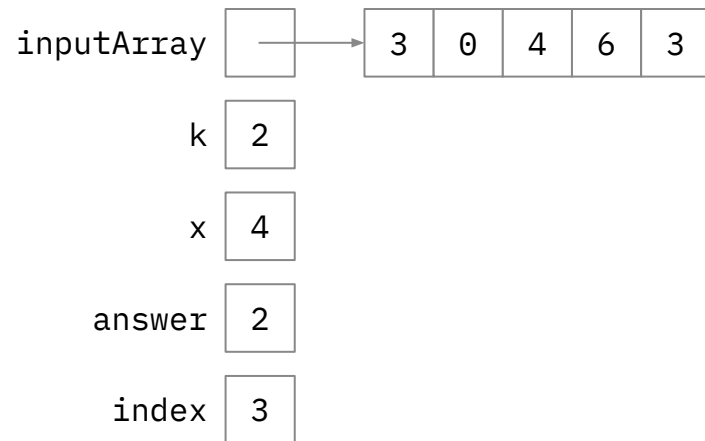
**What does this return when the input array is [3, 0, 4, 6, 3] and k is 2?
What does this answer mean?**

2 Reading Code: A Mystery

```

1  public static int mystery1(int[] inputArray, int k) {
2      int x = inputArray[k];
3      int answer = k;
4      int index = k + 1;
5      while (index < inputArray.length) { // True
6          if (inputArray[index] < x) {
7              x = inputArray[index];
8              answer = index;
9          }
10         index = index + 1;
11     }
12     return answer;
13 }

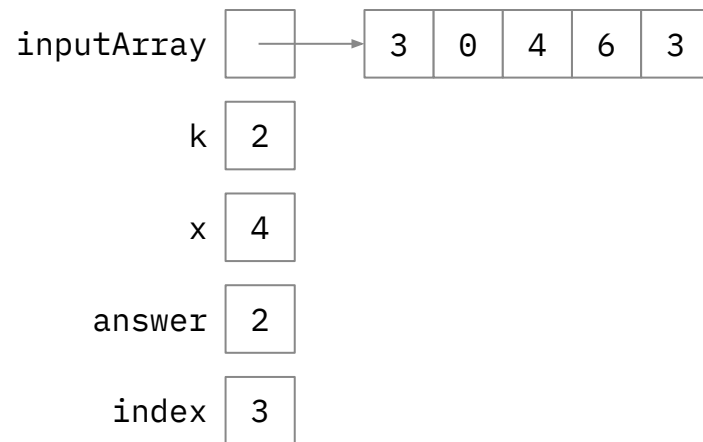
```



**What does this return when the input array is [3, 0, 4, 6, 3] and k is 2?
What does this answer mean?**

2 Reading Code: A Mystery

```
1 public static int mystery1(int[] inputArray, int k) {
2     int x = inputArray[k];
3     int answer = k;
4     int index = k + 1;
5     while (index < inputArray.length) {
6         if (inputArray[index] < x) { // False
7             x = inputArray[index];
8             answer = index;
9         }
10        index = index + 1;
11    }
12    return answer;
13 }
```



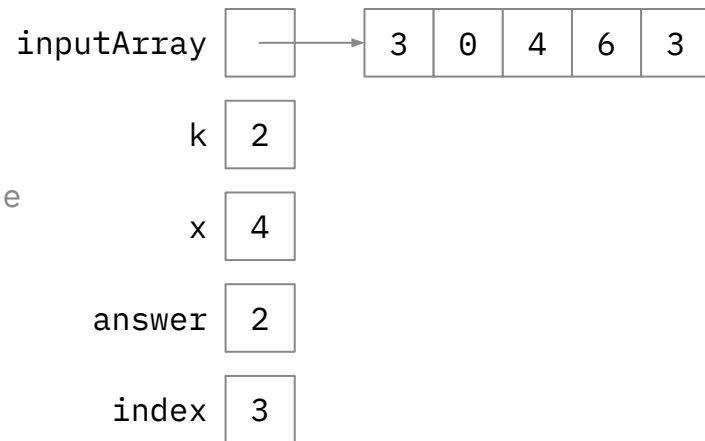
**What does this return when the input array is [3, 0, 4, 6, 3] and k is 2?
What does this answer mean?**

2 Reading Code: A Mystery

```

1  public static int mystery1(int[] inputArray, int k) {
2      int x = inputArray[k];
3      int answer = k;
4      int index = k + 1;
5      while (index < inputArray.length) {
6          if (inputArray[index] < x) {
7              x = inputArray[index]; // Skip this line
8              answer = index; // and this one
9          }
10         index = index + 1;
11     }
12     return answer;
13 }

```



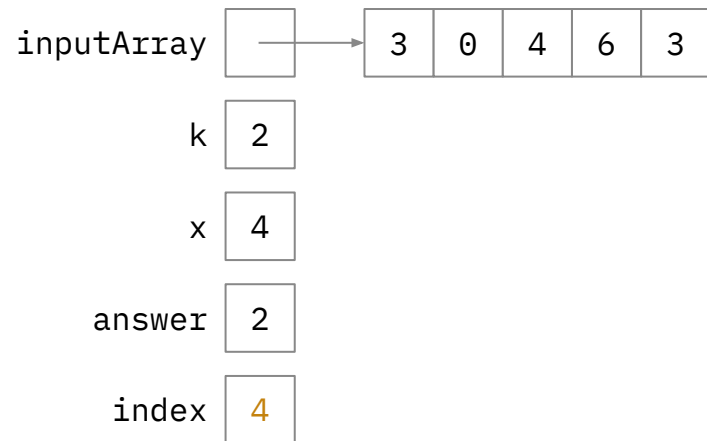
**What does this return when the input array is [3, 0, 4, 6, 3] and k is 2?
What does this answer mean?**

2 Reading Code: A Mystery

```

1  public static int mystery1(int[] inputArray, int k) {
2      int x = inputArray[k];
3      int answer = k;
4      int index = k + 1;
5      while (index < inputArray.length) {
6          if (inputArray[index] < x) {
7              x = inputArray[index];
8              answer = index;
9          }
10         index = index + 1;
11     }
12     return answer;
13 }

```



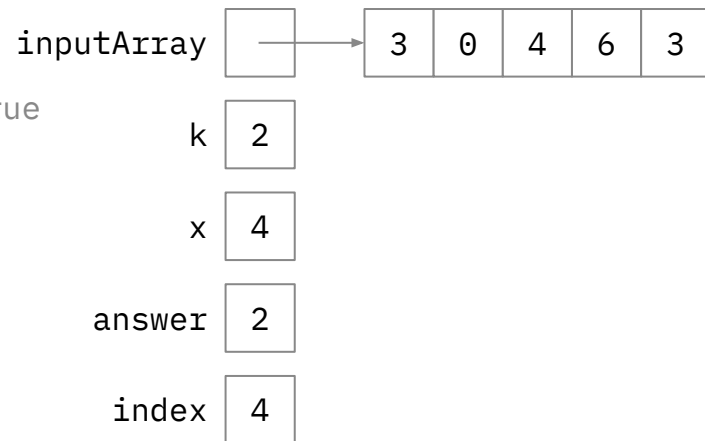
**What does this return when the input array is [3, 0, 4, 6, 3] and k is 2?
What does this answer mean?**

2 Reading Code: A Mystery

```

1  public static int mystery1(int[] inputArray, int k) {
2      int x = inputArray[k];
3      int answer = k;
4      int index = k + 1;
5      while (index < inputArray.length) { // still True
6          if (inputArray[index] < x) {
7              x = inputArray[index];
8              answer = index;
9          }
10         index = index + 1;
11     }
12     return answer;
13 }

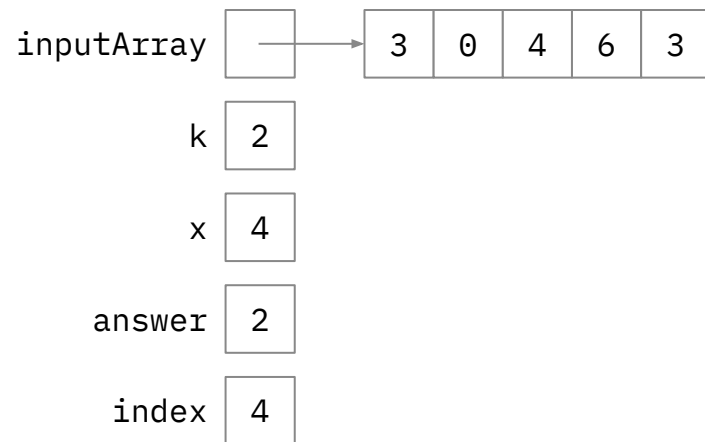
```



**What does this return when the input array is [3, 0, 4, 6, 3] and k is 2?
What does this answer mean?**

2 Reading Code: A Mystery

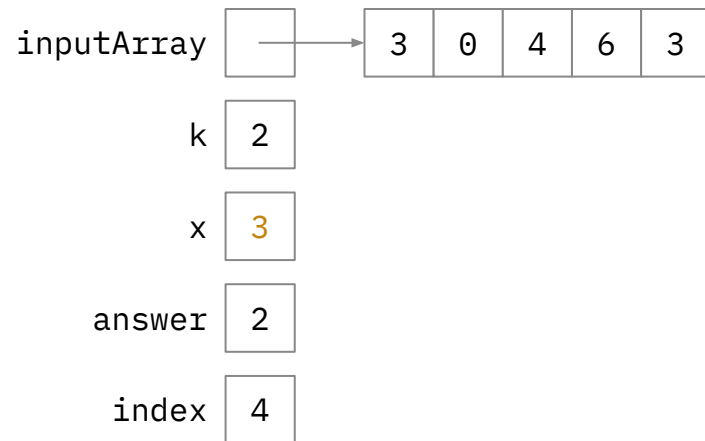
```
1 public static int mystery1(int[] inputArray, int k) {
2     int x = inputArray[k];
3     int answer = k;
4     int index = k + 1;
5     while (index < inputArray.length) {
6         if (inputArray[index] < x) { // True
7             x = inputArray[index];
8             answer = index;
9         }
10        index = index + 1;
11    }
12    return answer;
13 }
```



**What does this return when the input array is [3, 0, 4, 6, 3] and k is 2?
What does this answer mean?**

2 Reading Code: A Mystery

```
1 public static int mystery1(int[] inputArray, int k) {
2     int x = inputArray[k];
3     int answer = k;
4     int index = k + 1;
5     while (index < inputArray.length) {
6         if (inputArray[index] < x) {
7             x = inputArray[index];
8             answer = index;
9         }
10        index = index + 1;
11    }
12    return answer;
13 }
```



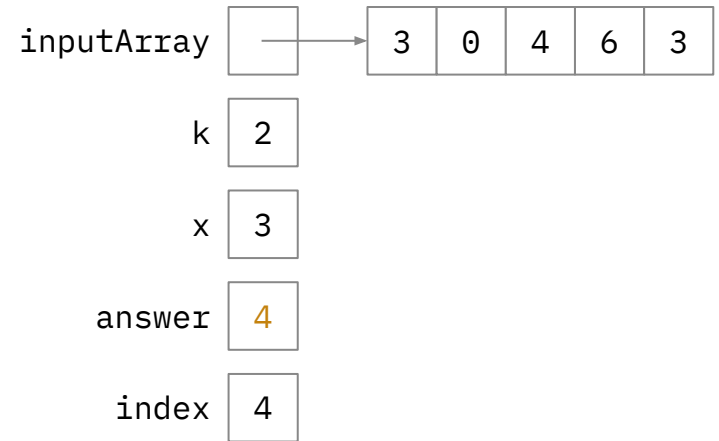
**What does this return when the input array is [3, 0, 4, 6, 3] and k is 2?
What does this answer mean?**

2 Reading Code: A Mystery

```

1  public static int mystery1(int[] inputArray, int k) {
2      int x = inputArray[k];
3      int answer = k;
4      int index = k + 1;
5      while (index < inputArray.length) {
6          if (inputArray[index] < x) {
7              x = inputArray[index];
8              answer = index;
9          }
10         index = index + 1;
11     }
12     return answer;
13 }

```



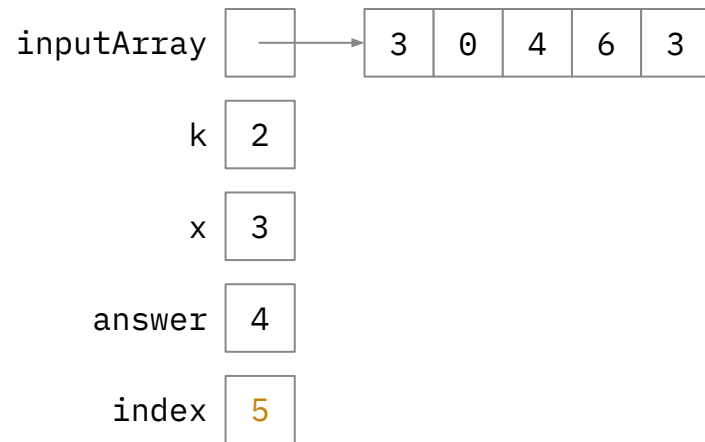
**What does this return when the input array is [3, 0, 4, 6, 3] and k is 2?
What does this answer mean?**

2 Reading Code: A Mystery

```

1  public static int mystery1(int[] inputArray, int k) {
2      int x = inputArray[k];
3      int answer = k;
4      int index = k + 1;
5      while (index < inputArray.length) {
6          if (inputArray[index] < x) {
7              x = inputArray[index];
8              answer = index;
9          }
10         index = index + 1;
11     }
12     return answer;
13 }

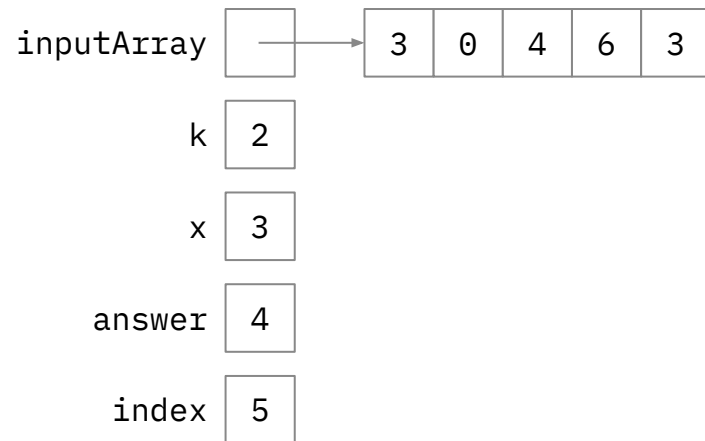
```



**What does this return when the input array is [3, 0, 4, 6, 3] and k is 2?
What does this answer mean?**

2 Reading Code: A Mystery

```
1 public static int mystery1(int[] inputArray, int k) {
2     int x = inputArray[k];
3     int answer = k;
4     int index = k + 1;
5     while (index < inputArray.length) { // False
6         if (inputArray[index] < x) {
7             x = inputArray[index];
8             answer = index;
9         }
10        index = index + 1;
11    }
12    return answer;
13 }
```



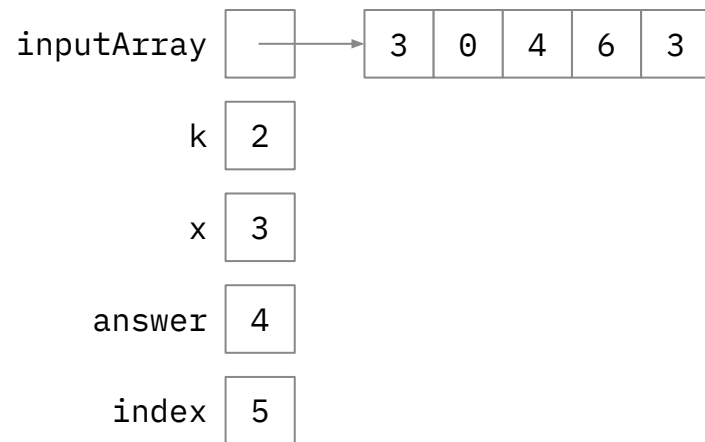
**What does this return when the input array is [3, 0, 4, 6, 3] and k is 2?
What does this answer mean?**

2 Reading Code: A Mystery

```

1  public static int mystery1(int[] inputArray, int k) {
2      int x = inputArray[k];
3      int answer = k;
4      int index = k + 1;
5      while (index < inputArray.length) {
6          if (inputArray[index] < x) { // Skip
7              x = inputArray[index]; // all
8              answer = index; // of
9          } // these
10         index = index + 1; // lines
11     }
12     return answer;
13 }

```



**What does this return when the input array is [3, 0, 4, 6, 3] and k is 2?
What does this answer mean?**

2 Reading Code: A Mystery

```
1 public static int mystery1(int[] inputArray, int k) {
2     int x = inputArray[k];
3     int answer = k;
4     int index = k + 1;
5     while (index < inputArray.length) {
6         if (inputArray[index] < x) {
7             x = inputArray[index];
8             answer = index;
9         }
10        index = index + 1;
11    }
12    return answer;
13 }
```

Return: 4

The returned value is the index of the smallest value in the array that occurs at or after index k.

What does this return when the input array is [3, 0, 4, 6, 3] and k is 2?

What does this answer mean?

2 Reading Code: A Mystery *Extra*

```
1 public static void mystery2(int[] inputArray) {
2     int index = 0;
3     while (index < inputArray.length) {
4         int targetIndex = mystery1(inputArray, index);
5         int temp = inputArray[targetIndex];
6         inputarray[targetIndex] = inputArray[index];
7         inputArray[index] = temp;
8         index = index + 1;
9     }
10 }
```

What does this return when the input array is [3, 0, 4, 6, 3]?
What does this function do?

2 Reading Code: A Mystery *Extra*

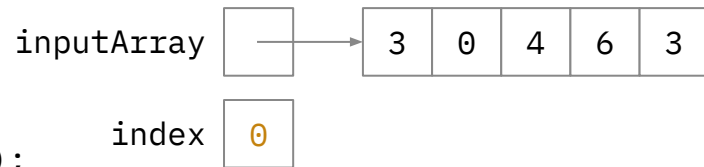
```
1 public static void mystery2(int[] inputArray) {
2     int index = 0;
3     while (index < inputArray.length) {
4         int targetIndex = mystery1(inputArray, index);
5         int temp = inputArray[targetIndex];
6         inputarray[targetIndex] = inputArray[index];
7         inputArray[index] = temp;
8         index = index + 1;
9     }
10 }
```



What does this return when the input array is [3, 0, 4, 6, 3]?
What does this function do?

2 Reading Code: A Mystery *Extra*

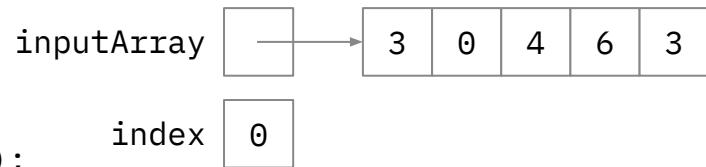
```
1 public static void mystery2(int[] inputArray) {  
2     int index = 0;  
3     while (index < inputArray.length) {  
4         int targetIndex = mystery1(inputArray, index);  
5         int temp = inputArray[targetIndex];  
6         inputarray[targetIndex] = inputArray[index];  
7         inputArray[index] = temp;  
8         index = index + 1;  
9     }  
10 }
```



What does this return when the input array is [3, 0, 4, 6, 3]?
What does this function do?

2 Reading Code: A Mystery *Extra*

```
1 public static void mystery2(int[] inputArray) {
2     int index = 0;
3     while (index < inputArray.length) { // True
4         int targetIndex = mystery1(inputArray, index);
5         int temp = inputArray[targetIndex];
6         inputarray[targetIndex] = inputArray[index];
7         inputArray[index] = temp;
8         index = index + 1;
9     }
10 }
```



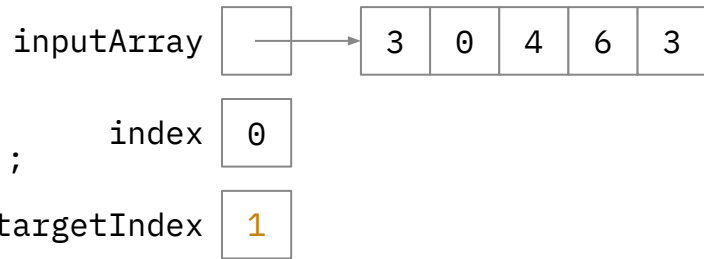
What does this return when the input array is [3, 0, 4, 6, 3]?
What does this function do?

2 Reading Code: A Mystery *Extra*

```

1 public static void mystery2(int[] inputArray) {
2     int index = 0;
3     while (index < inputArray.length) {
4         int targetIndex = mystery1(inputArray, index);
5         int temp = inputArray[targetIndex];
6         inputarray[targetIndex] = inputArray[index];
7         inputArray[index] = temp;
8         index = index + 1;
9     }
10 }

```



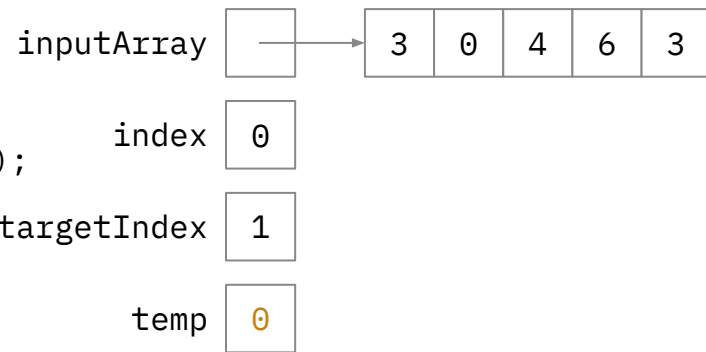
What does this return when the input array is [3, 0, 4, 6, 3]?
What does this function do?

2 Reading Code: A Mystery *Extra*

```

1 public static void mystery2(int[] inputArray) {
2     int index = 0;
3     while (index < inputArray.length) {
4         int targetIndex = mystery1(inputArray, index);
5         int temp = inputArray[targetIndex];
6         inputarray[targetIndex] = inputArray[index];
7         inputArray[index] = temp;
8         index = index + 1;
9     }
10 }

```

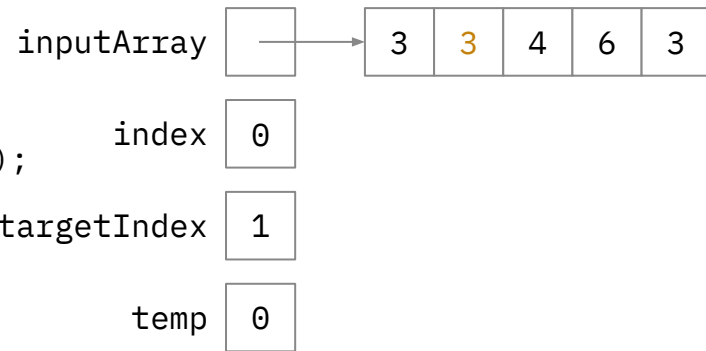


What does this return when the input array is [3, 0, 4, 6, 3]?
What does this function do?

2 Reading Code: A Mystery *Extra*

```

1  public static void mystery2(int[] inputArray) {
2      int index = 0;
3      while (index < inputArray.length) {
4          int targetIndex = mystery1(inputArray, index);
5          int temp = inputArray[targetIndex];
6          inputarray[targetIndex] = inputArray[index];
7          inputArray[index] = temp;
8          index = index + 1;
9      }
10 }
```



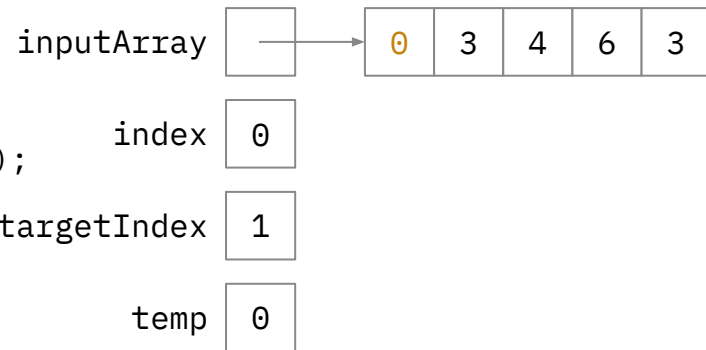
What does this return when the input array is [3, 0, 4, 6, 3]?
What does this function do?

2 Reading Code: A Mystery *Extra*

```

1 public static void mystery2(int[] inputArray) {
2     int index = 0;
3     while (index < inputArray.length) {
4         int targetIndex = mystery1(inputArray, index);
5         int temp = inputArray[targetIndex];
6         inputarray[targetIndex] = inputArray[index];
7         inputArray[index] = temp;
8         index = index + 1;
9     }
10 }

```



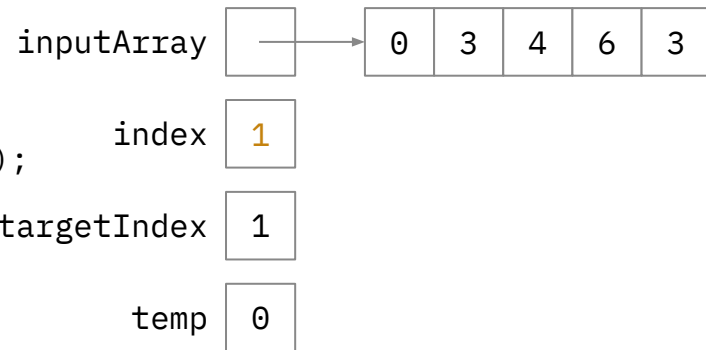
What does this return when the input array is [3, 0, 4, 6, 3]?
What does this function do?

2 Reading Code: A Mystery *Extra*

```

1 public static void mystery2(int[] inputArray) {
2     int index = 0;
3     while (index < inputArray.length) {
4         int targetIndex = mystery1(inputArray, index);
5         int temp = inputArray[targetIndex];
6         inputarray[targetIndex] = inputArray[index];
7         inputArray[index] = temp;
8         index = index + 1;
9     }
10 }

```

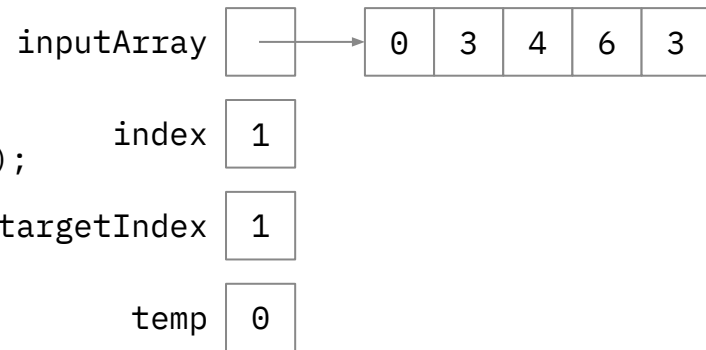


What does this return when the input array is [3, 0, 4, 6, 3]?
What does this function do?

2 Reading Code: A Mystery *Extra*

```

1 public static void mystery2(int[] inputArray) {
2     int index = 0;
3     while (index < inputArray.length) { // True
4         int targetIndex = mystery1(inputArray, index);
5         int temp = inputArray[targetIndex];
6         inputarray[targetIndex] = inputArray[index];
7         inputArray[index] = temp;
8         index = index + 1;
9     }
10 }
```



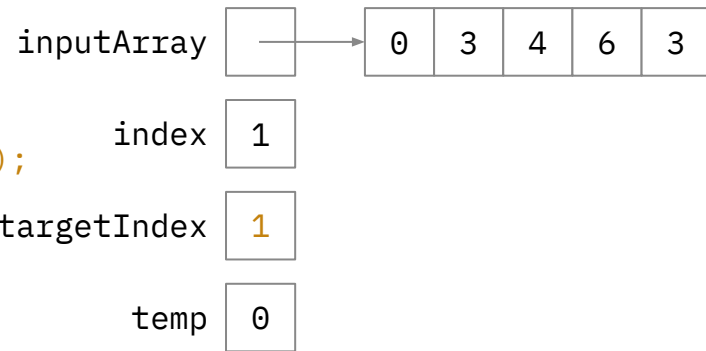
What does this return when the input array is [3, 0, 4, 6, 3]?
What does this function do?

2 Reading Code: A Mystery *Extra*

```

1 public static void mystery2(int[] inputArray) {
2     int index = 0;
3     while (index < inputArray.length) {
4         int targetIndex = mystery1(inputArray, index);
5         int temp = inputArray[targetIndex];
6         inputarray[targetIndex] = inputArray[index];
7         inputArray[index] = temp;
8         index = index + 1;
9     }
10 }

```

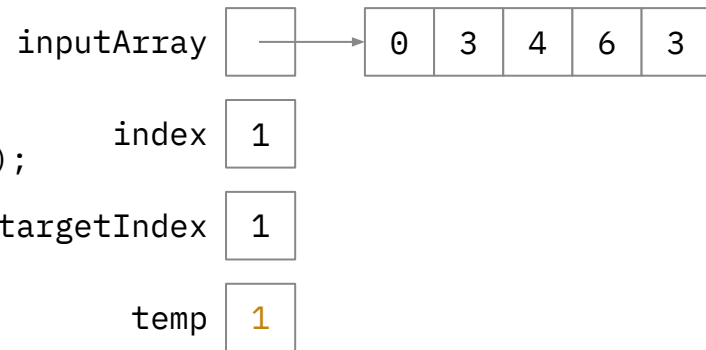


What does this return when the input array is [3, 0, 4, 6, 3]?
What does this function do?

2 Reading Code: A Mystery *Extra*

```

1 public static void mystery2(int[] inputArray) {
2     int index = 0;
3     while (index < inputArray.length) {
4         int targetIndex = mystery1(inputArray, index);
5         int temp = inputArray[targetIndex];
6         inputarray[targetIndex] = inputArray[index];
7         inputArray[index] = temp;
8         index = index + 1;
9     }
10 }
```



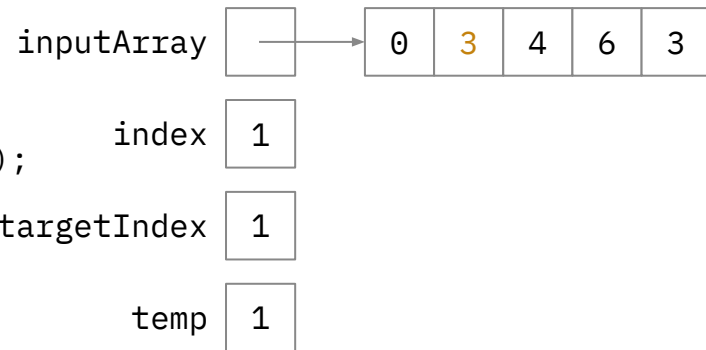
What does this return when the input array is [3, 0, 4, 6, 3]?
What does this function do?

2 Reading Code: A Mystery *Extra*

```

1 public static void mystery2(int[] inputArray) {
2     int index = 0;
3     while (index < inputArray.length) {
4         int targetIndex = mystery1(inputArray, index);
5         int temp = inputArray[targetIndex];
6         inputarray[targetIndex] = inputArray[index];
7         inputArray[index] = temp;
8         index = index + 1;
9     }
10 }

```



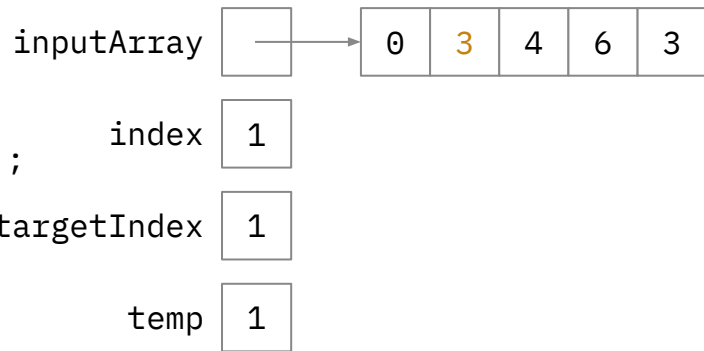
What does this return when the input array is [3, 0, 4, 6, 3]?
What does this function do?

2 Reading Code: A Mystery *Extra*

```

1  public static void mystery2(int[] inputArray) {
2      int index = 0;
3      while (index < inputArray.length) {
4          int targetIndex = mystery1(inputArray, index);
5          int temp = inputArray[targetIndex];
6          inputarray[targetIndex] = inputArray[index];
7          inputArray[index] = temp;
8          index = index + 1;
9      }
10 }

```

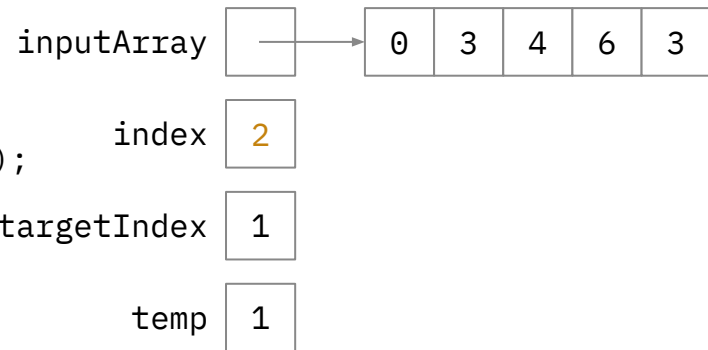


What does this return when the input array is [3, 0, 4, 6, 3]?
What does this function do?

2 Reading Code: A Mystery *Extra*

```

1 public static void mystery2(int[] inputArray) {
2     int index = 0;
3     while (index < inputArray.length) {
4         int targetIndex = mystery1(inputArray, index);
5         int temp = inputArray[targetIndex];
6         inputarray[targetIndex] = inputArray[index];
7         inputArray[index] = temp;
8         index = index + 1;
9     }
10 }
```



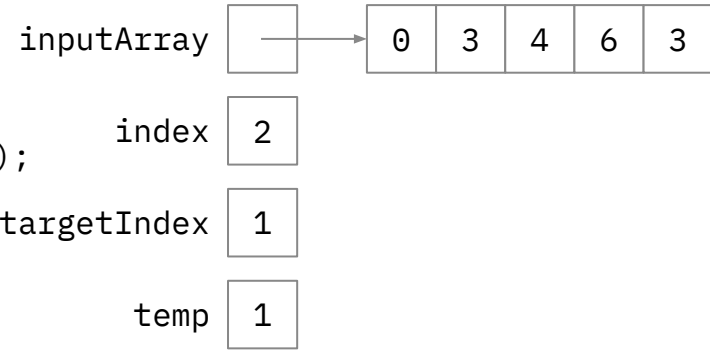
What does this return when the input array is [3, 0, 4, 6, 3]?
What does this function do?

2 Reading Code: A Mystery *Extra*

```

1  public static void mystery2(int[] inputArray) {
2      int index = 0;
3      while (index < inputArray.length) { // True
4          int targetIndex = mystery1(inputArray, index);
5          int temp = inputArray[targetIndex];
6          inputarray[targetIndex] = inputArray[index];
7          inputArray[index] = temp;
8          index = index + 1;
9      }
10 }

```



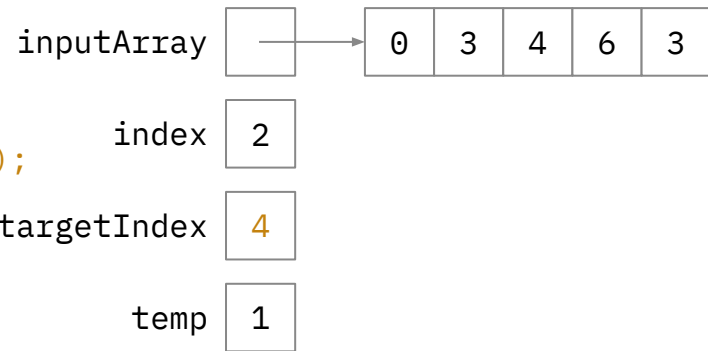
What does this return when the input array is [3, 0, 4, 6, 3]?
What does this function do?

2 Reading Code: A Mystery *Extra*

```

1 public static void mystery2(int[] inputArray) {
2     int index = 0;
3     while (index < inputArray.length) {
4         int targetIndex = mystery1(inputArray, index);
5         int temp = inputArray[targetIndex];
6         inputarray[targetIndex] = inputArray[index];
7         inputArray[index] = temp;
8         index = index + 1;
9     }
10 }

```




What does this return when the input array is [3, 0, 4, 6, 3]?
What does this function do?


2 Reading Code: A Mystery *Extra*


```


1 public static void mystery2(int[] inputArray) {
2     int index = 0;
3     while (index < inputArray.length) {
4         int targetIndex = mystery1(inputArray, index);
5         int temp = inputArray[targetIndex];
6         inputarray[targetIndex] = inputArray[index];
7         inputArray[index] = temp;
8         index = index + 1;
9     }
10 }

```

inputArray 

index 

targetIndex 

temp 

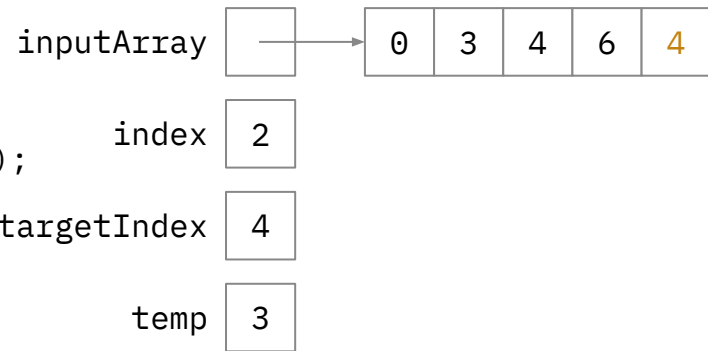
What does this return when the input array is [3, 0, 4, 6, 3]?
What does this function do?

2 Reading Code: A Mystery *Extra*

```

1  public static void mystery2(int[] inputArray) {
2      int index = 0;
3      while (index < inputArray.length) {
4          int targetIndex = mystery1(inputArray, index);
5          int temp = inputArray[targetIndex];
6          inputarray[targetIndex] = inputArray[index];
7          inputArray[index] = temp;
8          index = index + 1;
9      }
10 }

```



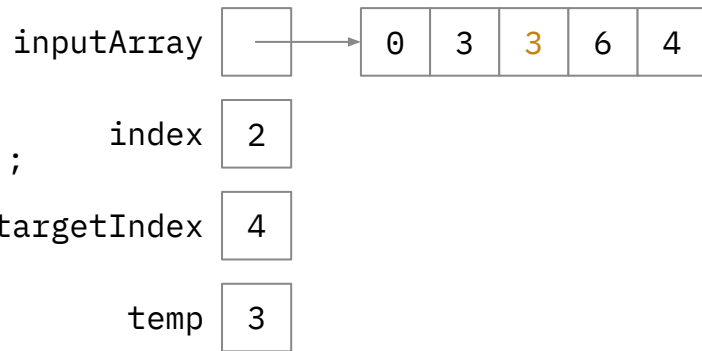
What does this return when the input array is [3, 0, 4, 6, 3]?
What does this function do?

2 Reading Code: A Mystery *Extra*

```

1  public static void mystery2(int[] inputArray) {
2      int index = 0;
3      while (index < inputArray.length) {
4          int targetIndex = mystery1(inputArray, index);
5          int temp = inputArray[targetIndex];
6          inputarray[targetIndex] = inputArray[index];
7          inputArray[index] = temp;
8          index = index + 1;
9      }
10 }

```



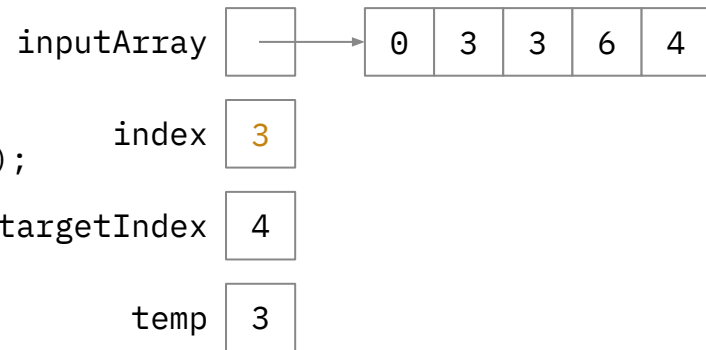
What does this return when the input array is [3, 0, 4, 6, 3]?
What does this function do?

2 Reading Code: A Mystery *Extra*

```

1 public static void mystery2(int[] inputArray) {
2     int index = 0;
3     while (index < inputArray.length) {
4         int targetIndex = mystery1(inputArray, index);
5         int temp = inputArray[targetIndex];
6         inputarray[targetIndex] = inputArray[index];
7         inputArray[index] = temp;
8         index = index + 1;
9     }
10 }

```



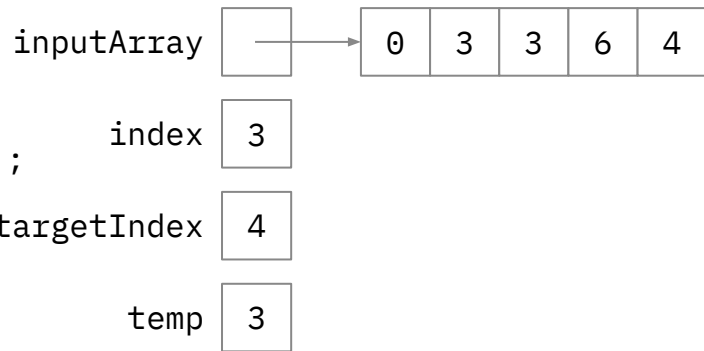
What does this return when the input array is [3, 0, 4, 6, 3]?
What does this function do?

2 Reading Code: A Mystery *Extra*

```

1 public static void mystery2(int[] inputArray) {
2     int index = 0;
3     while (index < inputArray.length) { // True
4         int targetIndex = mystery1(inputArray, index);
5         int temp = inputArray[targetIndex];
6         inputarray[targetIndex] = inputArray[index];
7         inputArray[index] = temp;
8         index = index + 1;
9     }
10 }

```



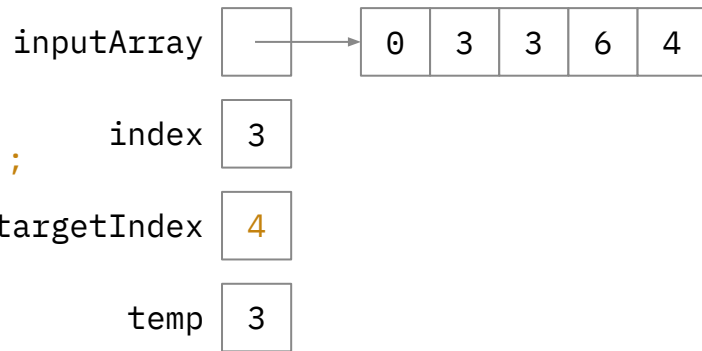
What does this return when the input array is [3, 0, 4, 6, 3]?
What does this function do?

2 Reading Code: A Mystery *Extra*

```

1 public static void mystery2(int[] inputArray) {
2     int index = 0;
3     while (index < inputArray.length) {
4         int targetIndex = mystery1(inputArray, index);
5         int temp = inputArray[targetIndex];
6         inputarray[targetIndex] = inputArray[index];
7         inputArray[index] = temp;
8         index = index + 1;
9     }
10 }

```



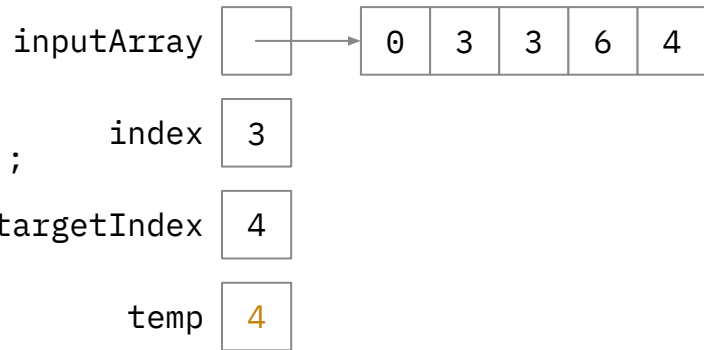
What does this return when the input array is [3, 0, 4, 6, 3]?
What does this function do?

2 Reading Code: A Mystery *Extra*

```

1  public static void mystery2(int[] inputArray) {
2      int index = 0;
3      while (index < inputArray.length) {
4          int targetIndex = mystery1(inputArray, index);
5          int temp = inputArray[targetIndex];
6          inputarray[targetIndex] = inputArray[index];
7          inputArray[index] = temp;
8          index = index + 1;
9      }
10 }

```



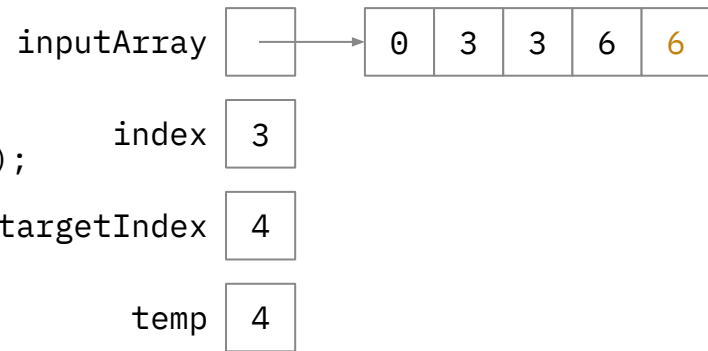
What does this return when the input array is [3, 0, 4, 6, 3]?
What does this function do?

2 Reading Code: A Mystery *Extra*

```

1  public static void mystery2(int[] inputArray) {
2      int index = 0;
3      while (index < inputArray.length) {
4          int targetIndex = mystery1(inputArray, index);
5          int temp = inputArray[targetIndex];
6          inputarray[targetIndex] = inputArray[index];
7          inputArray[index] = temp;
8          index = index + 1;
9      }
10 }

```




What does this return when the input array is [3, 0, 4, 6, 3]?
What does this function do?


2 Reading Code: A Mystery *Extra*


```


1 public static void mystery2(int[] inputArray) {
2     int index = 0;
3     while (index < inputArray.length) {
4         int targetIndex = mystery1(inputArray, index);
5         int temp = inputArray[targetIndex];
6         inputarray[targetIndex] = inputArray[index];
7         inputArray[index] = temp;
8         index = index + 1;
9     }
10 }

```

inputArray 

 index 

 targetIndex 

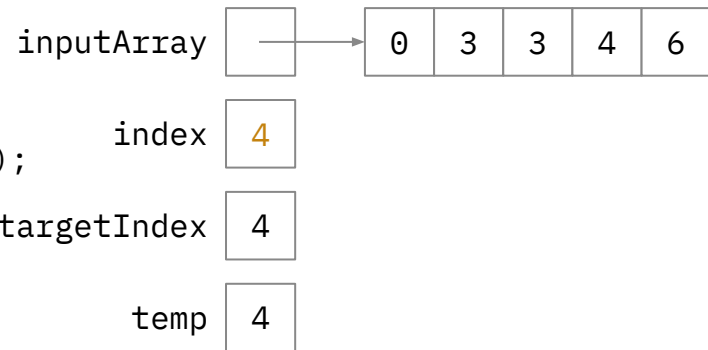
 temp 

What does this return when the input array is [3, 0, 4, 6, 3]?
What does this function do?

2 Reading Code: A Mystery *Extra*

```

1 public static void mystery2(int[] inputArray) {
2     int index = 0;
3     while (index < inputArray.length) {
4         int targetIndex = mystery1(inputArray, index);
5         int temp = inputArray[targetIndex];
6         inputarray[targetIndex] = inputArray[index];
7         inputArray[index] = temp;
8         index = index + 1;
9     }
10 }
```



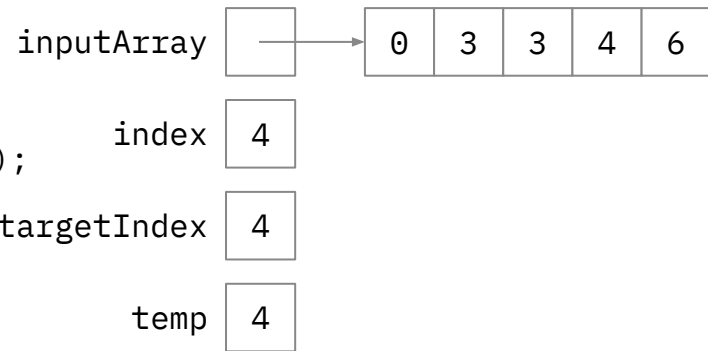
What does this return when the input array is [3, 0, 4, 6, 3]?
What does this function do?

2 Reading Code: A Mystery *Extra*

```

1 public static void mystery2(int[] inputArray) {
2     int index = 0;
3     while (index < inputArray.length) { // True
4         int targetIndex = mystery1(inputArray, index);
5         int temp = inputArray[targetIndex];
6         inputarray[targetIndex] = inputArray[index];
7         inputArray[index] = temp;
8         index = index + 1;
9     }
10 }

```



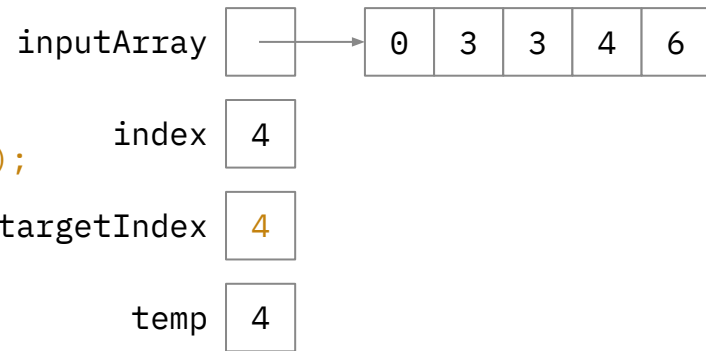
What does this return when the input array is [3, 0, 4, 6, 3]?
What does this function do?

2 Reading Code: A Mystery *Extra*

```

1 public static void mystery2(int[] inputArray) {
2     int index = 0;
3     while (index < inputArray.length) {
4         int targetIndex = mystery1(inputArray, index);
5         int temp = inputArray[targetIndex];
6         inputarray[targetIndex] = inputArray[index];
7         inputArray[index] = temp;
8         index = index + 1;
9     }
10 }

```




What does this return when the input array is [3, 0, 4, 6, 3]?
What does this function do?


2 Reading Code: A Mystery *Extra*


```


1 public static void mystery2(int[] inputArray) {
2     int index = 0;
3     while (index < inputArray.length) {
4         int targetIndex = mystery1(inputArray, index);
5         int temp = inputArray[targetIndex];
6         inputarray[targetIndex] = inputArray[index];
7         inputArray[index] = temp;
8         index = index + 1;
9     }
10 }

```

inputArray 

index 

targetIndex 

temp 

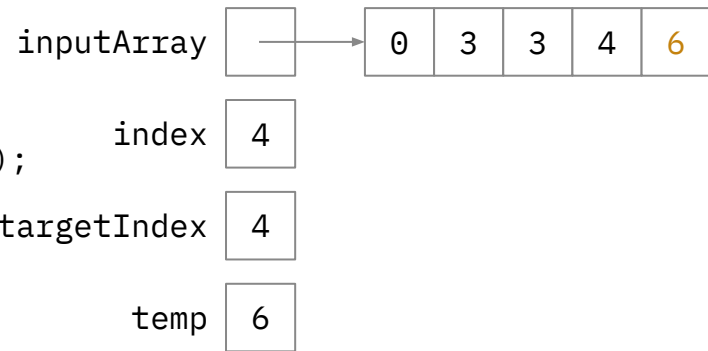
What does this return when the input array is [3, 0, 4, 6, 3]?
What does this function do?

2 Reading Code: A Mystery *Extra*

```

1 public static void mystery2(int[] inputArray) {
2     int index = 0;
3     while (index < inputArray.length) {
4         int targetIndex = mystery1(inputArray, index);
5         int temp = inputArray[targetIndex];
6         inputarray[targetIndex] = inputArray[index];
7         inputArray[index] = temp;
8         index = index + 1;
9     }
10 }

```



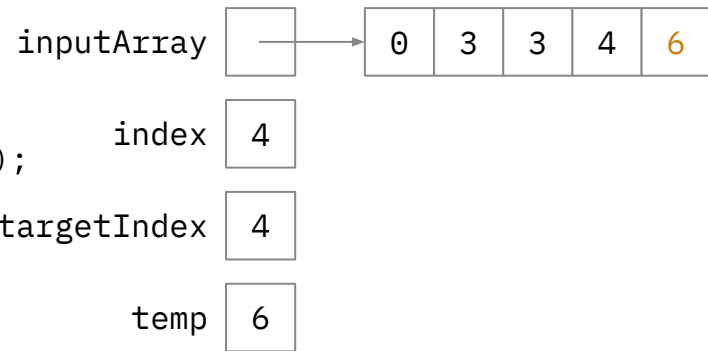
What does this return when the input array is [3, 0, 4, 6, 3]?
What does this function do?

2 Reading Code: A Mystery *Extra*

```

1 public static void mystery2(int[] inputArray) {
2     int index = 0;
3     while (index < inputArray.length) {
4         int targetIndex = mystery1(inputArray, index);
5         int temp = inputArray[targetIndex];
6         inputarray[targetIndex] = inputArray[index];
7         inputArray[index] = temp;
8         index = index + 1;
9     }
10 }

```



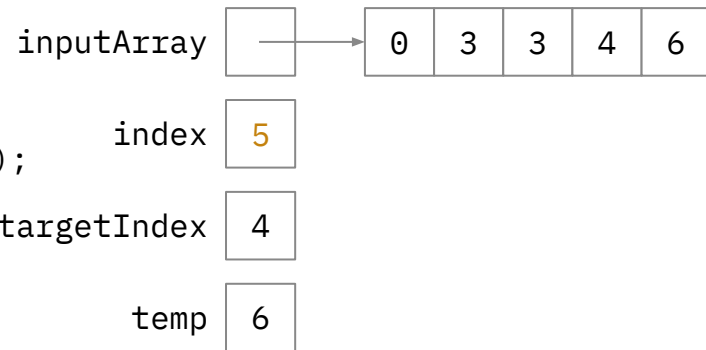
What does this return when the input array is [3, 0, 4, 6, 3]?
What does this function do?

2 Reading Code: A Mystery *Extra*

```

1 public static void mystery2(int[] inputArray) {
2     int index = 0;
3     while (index < inputArray.length) {
4         int targetIndex = mystery1(inputArray, index);
5         int temp = inputArray[targetIndex];
6         inputarray[targetIndex] = inputArray[index];
7         inputArray[index] = temp;
8         index = index + 1;
9     }
10 }

```



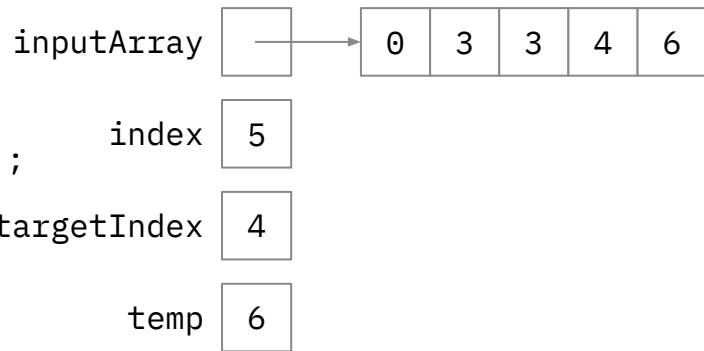
What does this return when the input array is [3, 0, 4, 6, 3]?
What does this function do?

2 Reading Code: A Mystery *Extra*

```

1 public static void mystery2(int[] inputArray) {
2     int index = 0;
3     while (index < inputArray.length) { // False
4         int targetIndex = mystery1(inputArray, index);
5         int temp = inputArray[targetIndex];
6         inputarray[targetIndex] = inputArray[index];
7         inputArray[index] = temp;
8         index = index + 1;
9     }
10 }

```



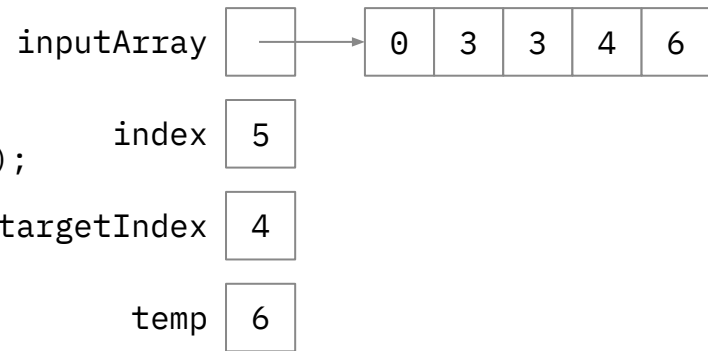
What does this return when the input array is [3, 0, 4, 6, 3]?
What does this function do?

2 Reading Code: A Mystery *Extra*

```

1 public static void mystery2(int[] inputArray) {
2     int index = 0;
3     while (index < inputArray.length) { // False
4         int targetIndex = mystery1(inputArray, index);
5         int temp = inputArray[targetIndex];
6         inputarray[targetIndex] = inputArray[index];
7         inputArray[index] = temp;
8         index = index + 1;
9     }
10 }

```



What does this return when the input array is [3, 0, 4, 6, 3]?
What does this function do?

2 Reading Code: A Mystery *Extra*

```
1 public static void mystery2(int[] inputArray) {
2     int index = 0;
3     while (index < inputArray.length) {
4         int targetIndex = mystery1(inputArray, index);
5         int temp = inputArray[targetIndex];
6         inputarray[targetIndex] = inputArray[index];
7         inputArray[index] = temp;
8         index = index + 1;
9     }
10 }
```

What does this return when the input array is [3, 0, 4, 6, 3]?
What does this function do?

Return: Nothing (return type is void)

This function sorts the array in increasing order in place, so our input array turns into [0, 3, 3, 4, 6]

3 Recursion Practice: Fibonacci

Implement a function `fib1` that recursively calculates the `N`th fibonacci number.

Hint: $\text{fib}(N) = \text{fib}(N-1) + \text{fib}(N-2)$

```
public static int fib1(int N) {
```

```
}
```


3 Recursion Practice: Fibonacci

Implement a function `fib1` that recursively calculates the `N`th fibonacci number.

Hint: $\text{fib}(N) = \text{fib}(N-1) + \text{fib}(N-2)$

```
public static int fib1(int N) {  
    if (N <= 1) { // Base case - can be written a few different ways  
        return N;  
    }  
  
}
```

3 Recursion Practice: Fibonacci

Implement a function `fib1` that recursively calculates the `N`th fibonacci number.

Hint: $\text{fib}(N) = \text{fib}(N-1) + \text{fib}(N-2)$

```
public static int fib1(int N) {
    if (N <= 1) {
        return N;
    } else {
        return fib1(N - 1) + fib1(N - 2);
    } // Just copying over the same recursive formula from the hint
}
```

3 Recursion Practice: Fibonacci *Extra*

Implement a function `fib2` that recursively calculates the `N`th fibonacci number more efficiently with new arguments `k`, `f0`, and `f1` in 5 lines or less.

Hint: To compute the `N`th fibonacci number, call `fib2(N, 0, 0 1)`

```
public static int fib2(int N, int k, int f0, int f1) {
```

```
}
```

3 Recursion Practice: Fibonacci *Extra*

Implement a function `fib2` that recursively calculates the `N`th fibonacci number more efficiently with new arguments `k`, `f0`, and `f1` in 5 lines or less.

Hint: To compute the `N`th fibonacci number, call `fib2(N, 0, 0 1)`

```
public static int fib2(int N, int k, int f0, int f1) {  
  
  
  
  
}  
/* N = the number of recursions in the end  
* k = the number recursions already accomplished (hint -> starts @ 0)  
* f0 = the second to last fib num (hint -> the 0th fibonacci number (0))  
* f1 = the last fib num (hint -> the 1st fibonacci number (1))  
*/
```

3 Recursion Practice: Fibonacci *Extra*

Implement a function `fib2` that recursively calculates the `N`th fibonacci number more efficiently with new arguments `k`, `f0`, and `f1` in 5 lines or less.

Hint: To compute the `N`th fibonacci number, call `fib2(N, 0, 0 1)`

```
public static int fib2(int N, int k, int f0, int f1) {  
    if (N == k) { // num recursions goal == num recursions accomplished  
        return f0; // when N == k, we want f0 (ex. N == 0, we want f0 = 0)  
    }  
  
}
```

3 Recursion Practice: Fibonacci *Extra*

Implement a function `fib2` that recursively calculates the `N`th fibonacci number more efficiently with new arguments `k`, `f0`, and `f1` in 5 lines or less.

Hint: To compute the `N`th fibonacci number, call `fib2(N, 0, 0 1)`

```
public static int fib2(int N, int k, int f0, int f1) {  
    if (N == k) {  
        return f0;  
    } else { // N stays same, k increments, f0 = f1, f1 = f1 + f2  
        return fib2(N, k + 1, f1, f0 + f1);  
    }  
}
```