More with Arrays

March 6, 2012

CMPSCI 121, Spring 2012

Introduction to Problem Solving with Computers

Prof. Learned-Miller

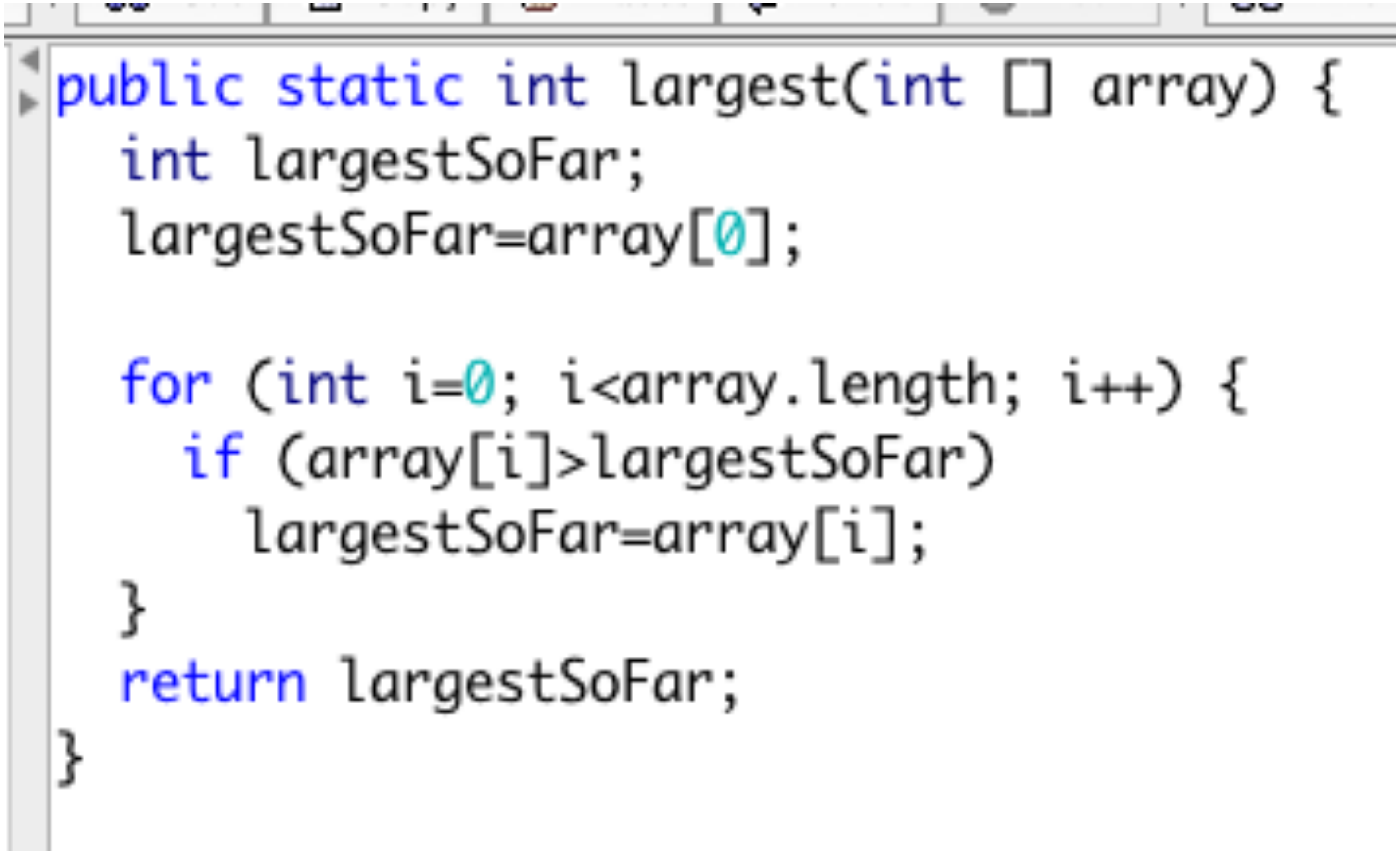
Algorithms

- Algorithm: *a sequence of steps or instructions guaranteed to solve a problem*.
  Heuristic: *a sequence of steps or instructions that provides a solution to a problem which may or may not be correct*.

Today

- Finding the largest number in an array.
- Finding the *index* of the largest number in an array.
- Swapping numbers in an array.
- Shifting the numbers in an array.
- Sorting the numbers in an array.

Largest number in array

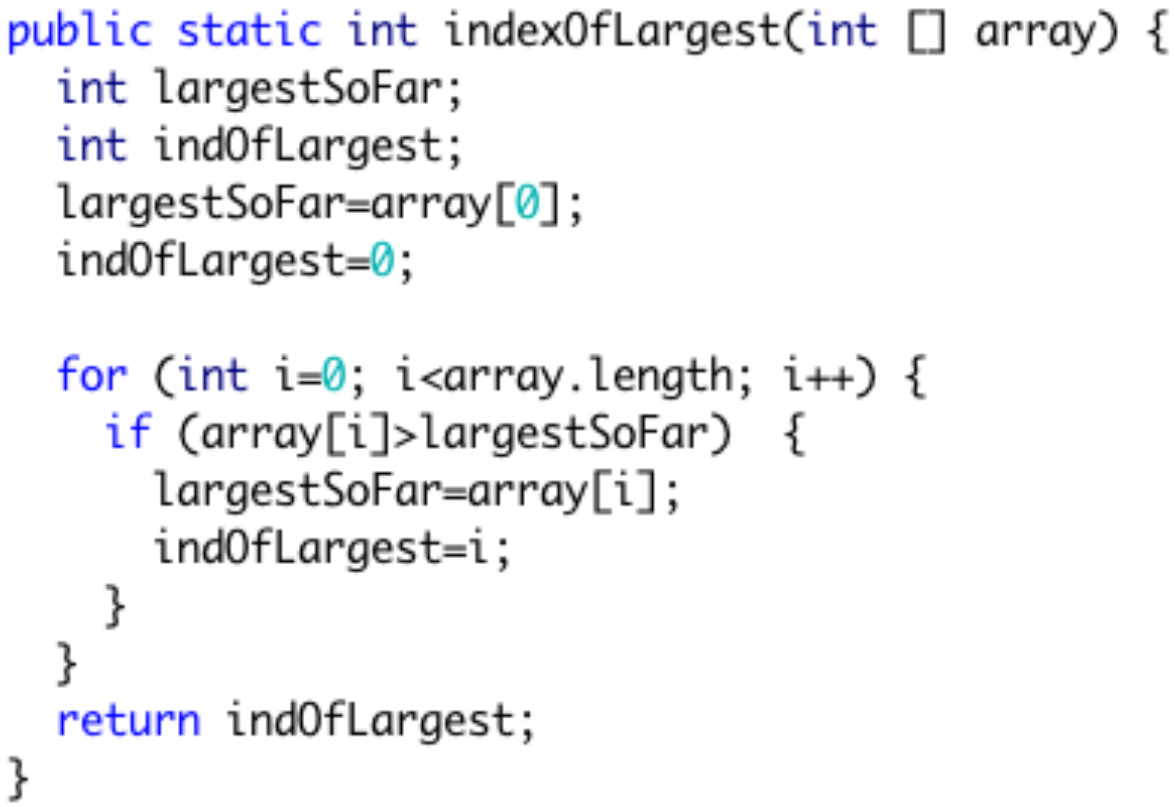


```
public static int largest(int [] array) {
  int largestSoFar;
  largestSoFar=array[0];

  for (int i=0; i<array.length; i++) {
    if (array[i]>largestSoFar)
      largestSoFar=array[i];
  }
  return largestSoFar;
}
```

Index of largest number

Index of largest number



```
public static int indexOfLargest(int [] array) {
  int largestSoFar;
  int indOfLargest;
  largestSoFar=array[0];
  indOfLargest=0;

  for (int i=0; i<array.length; i++) {
    if (array[i]>largestSoFar)  {
      largestSoFar=array[i];
      indOfLargest=i;
    }
  }
  return indOfLargest;
}
```

Swapping Values

- int x = 3;
- int y = 5;
- How to swap values of x and y?

Swapping Values

- int x = 3;
- int y = 5;
- How to swap values of x and y?
  - y=x;
  - x=y;

Swapping Values

- int x = 3;
- int y = 5;
- How to swap values of x and y?
  - ~~y=x;~~ // After this, x == 3, y == 3.
  - ~~x=y;~~ // After this, x == 3, y == 3.
  - // This is the wrong result!

Swapping Values

- int x = 3;
- int y = 5;
- How to swap values of x and y?
  - int temp = y;   // temp==5, x==3, y==5.
  - y = x;   // temp==5, x==3, y==3.
  - x = temp;   // temp==5, x==5, y==3.
  - // The correct result!

Shifting array elements

- Write a static method which takes an array as a parameter and shifts the elements in the array one place to the left. The first element of the array should be placed at the end of the array.

Starting point

7	3	8	2

Goal

3	8	2	7

```
public static void shiftForward(int [] array) {
```

Solution #1

Starting point

7	3	8	2

Goal

3	8	2	7

Solution #1

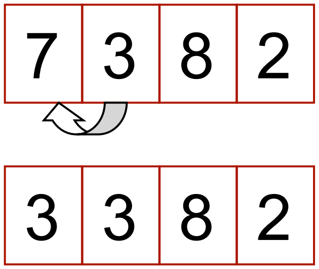
Starting point

7	3	8	2

Goal

3	8	2	7

Step 1?



7	3	8	2

3	3	8	2

Solution #1

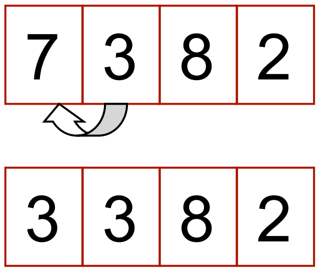
Starting point

7	3	8	2

Goal

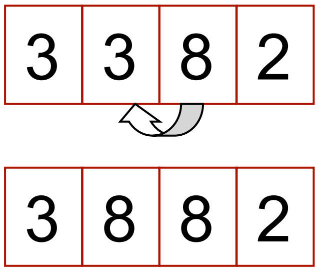
3	8	2	7

Step 1?



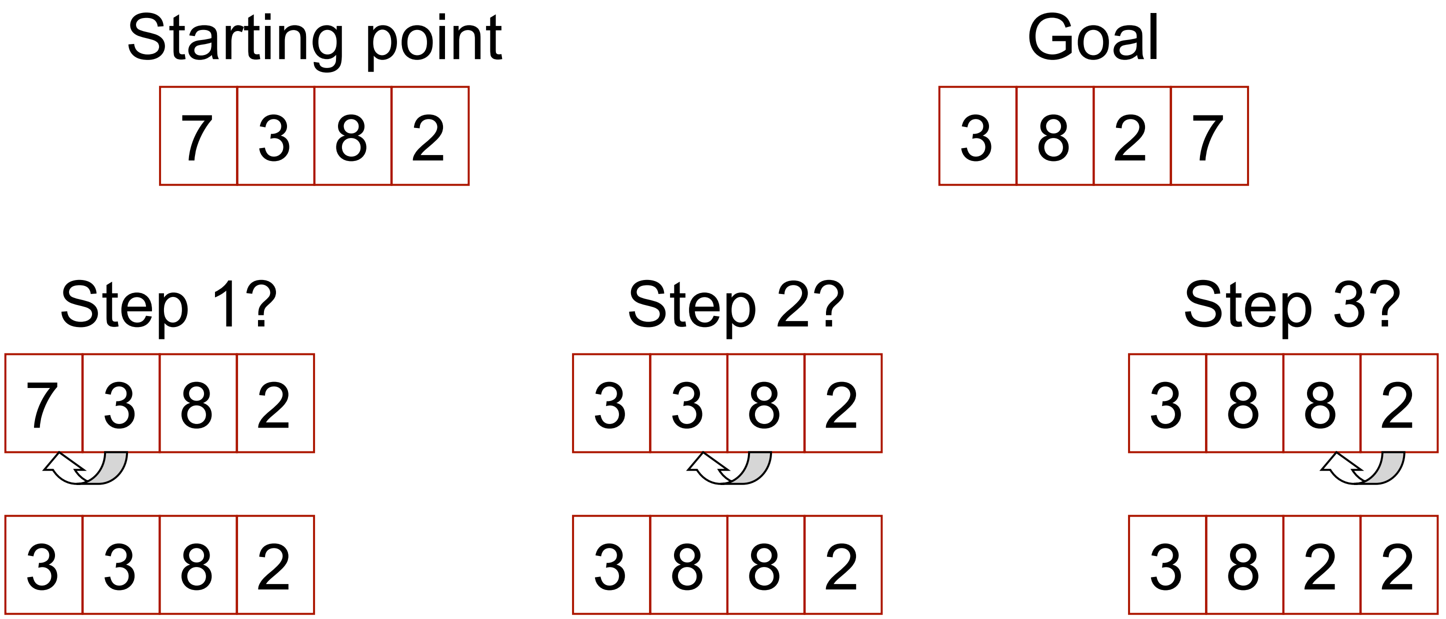
7	3	8	2
3	3	8	2

Step 2?



3	3	8	2
3	8	8	2

Solution #1



Starting point

7	3	8	2

Goal

3	8	2	7

Step 1?

7	3	8	2

3	3	8	2

Step 2?

3	3	8	2

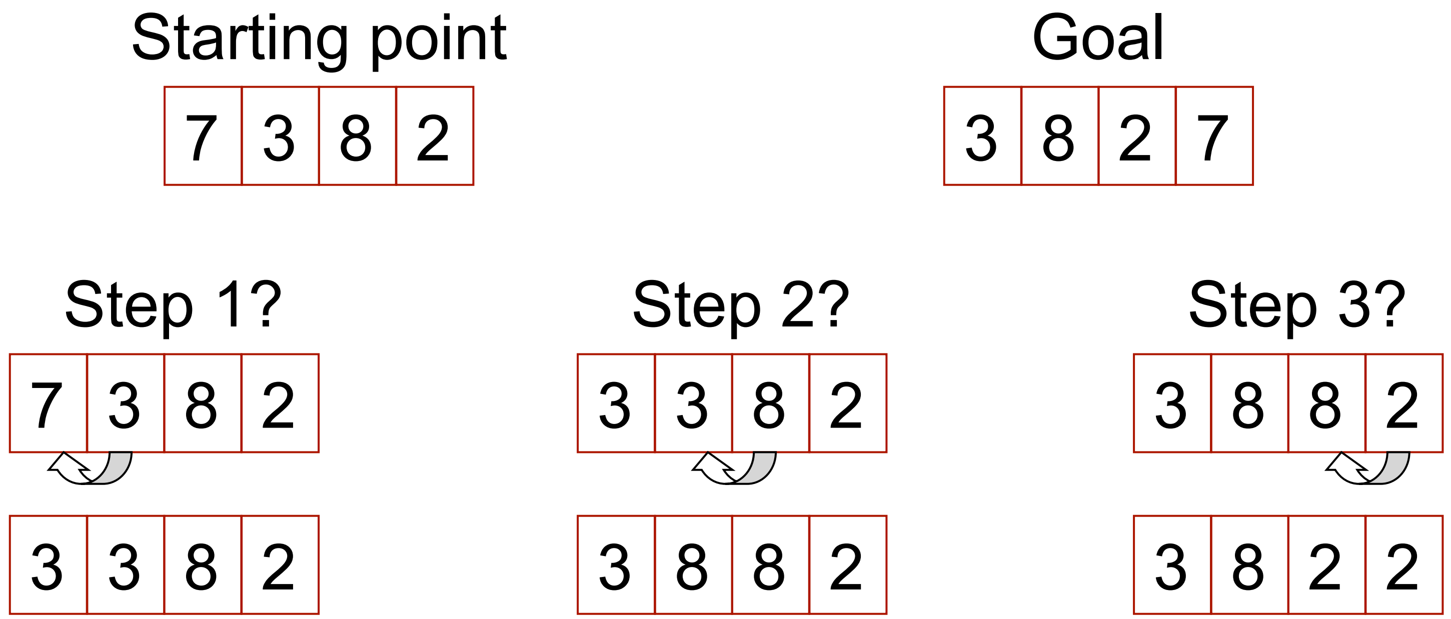
3	8	8	2

Step 3?

3	8	8	2

3	8	2	2

Solution #1



Starting point

7	3	8	2

Goal

3	8	2	7

Step 1?

7	3	8	2
3	3	8	2

Step 2?

3	3	8	2
3	8	8	2

Step 3?

3	8	8	2
3	8	2	2

We’ve almost got what we want!

Solution #1

Starting point

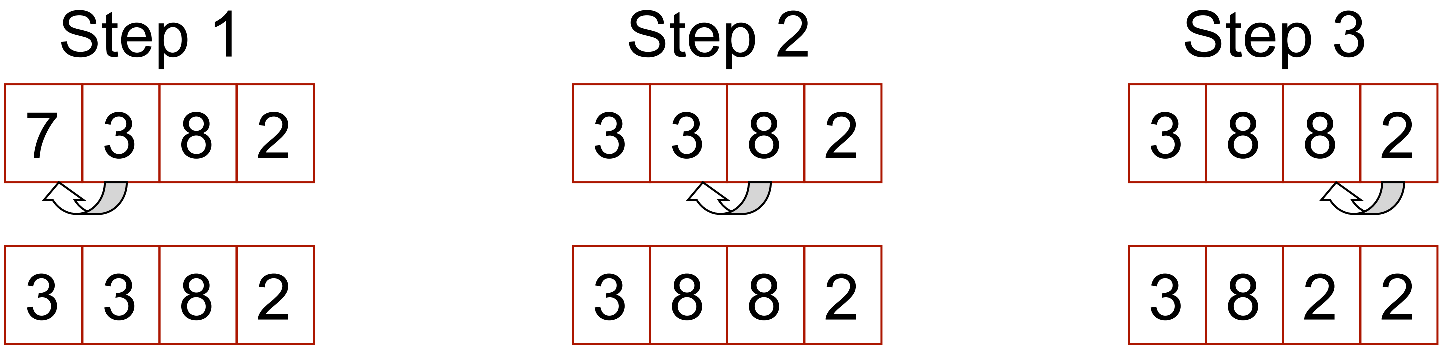
7	3	8	2

Goal

3	8	2	7

Step 0: Save the first value before we overwrite it!

```
int temp = array[0];
```



Step 1

7	3	8	2
3	3	8	2

Step 2

3	3	8	2
3	8	8	2

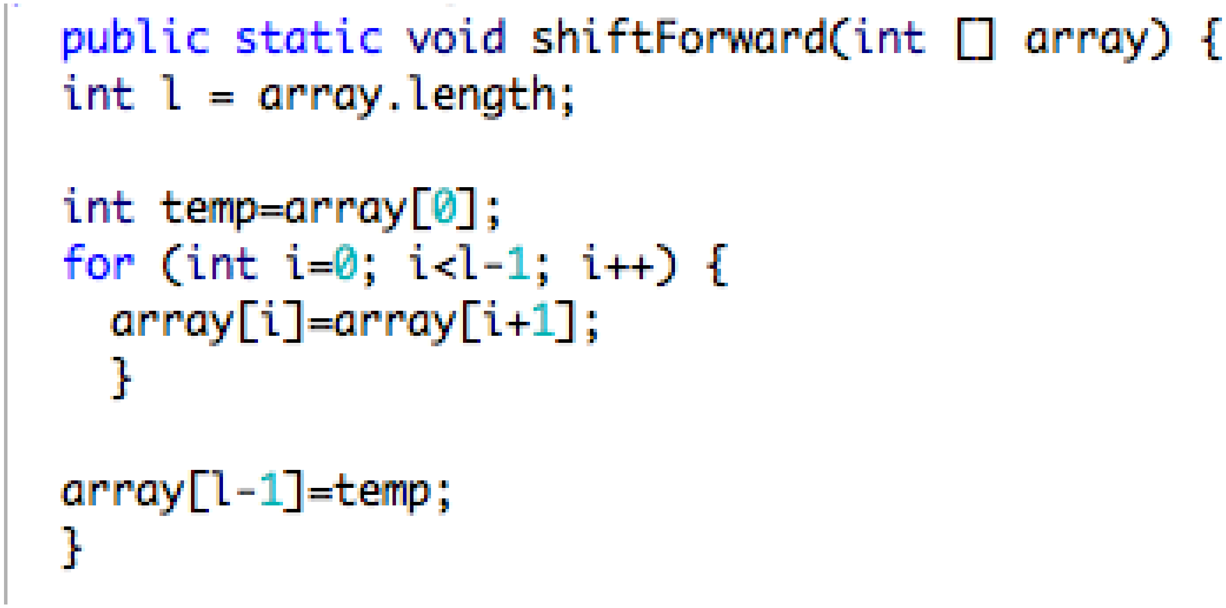
Step 3

3	8	8	2
3	8	2	2

```
array[array.length-1] = temp;
```

3	8	2	7

Solution #1
“In place solution”



```
public static void shiftForward(int [] array) {
int l = array.length;

int temp=array[0];
for (int i=0; i<l-1; i++) {
  array[i]=array[i+1];
  }

array[l-1]=temp;
}
```

Solution #2

Starting point

array	7	3	8	2

Goal

array	3	8	2	7

Solution #2

Starting point

array	7	3	8	2

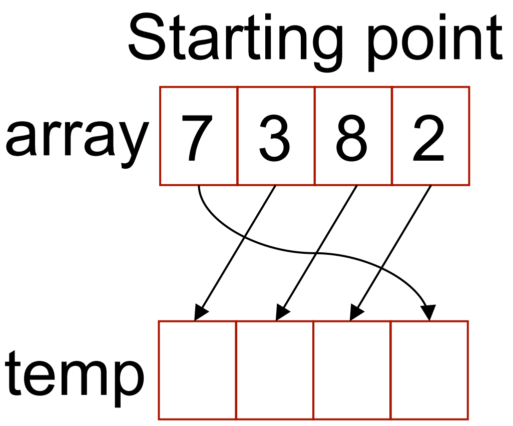
Goal

array	3	8	2	7

temp				

Temporary variable
“workspace”

Solution #2



Starting point

array	7	3	8	2
temp				

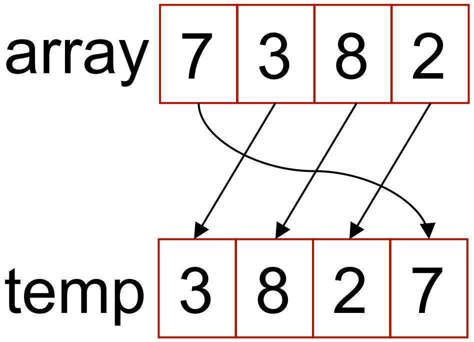
Temporary variable
“workspace”

Goal

array	3	8	2	7

Solution #2

Starting point



array 7 3 8 2

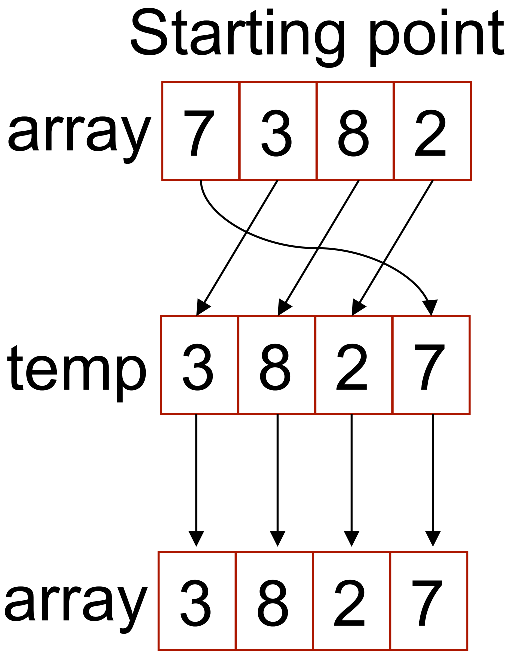
temp 3 8 2 7

Temporary variable
“workspace”

Goal

array 3 8 2 7

Solution #2



Starting point

array | 7 | 3 | 8 | 2

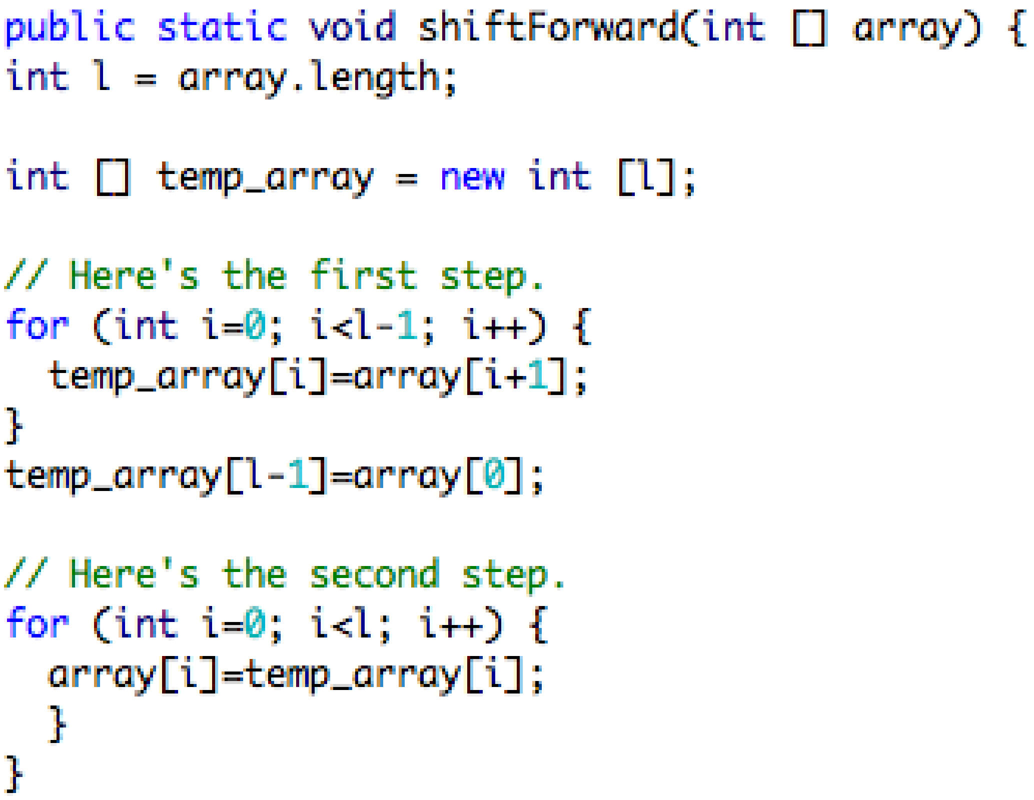
temp | 3 | 8 | 2 | 7

array | 3 | 8 | 2 | 7

Goal

array | 3 | 8 | 2 | 7

Solution #2
“Workspace solution”



```
public static void shiftForward(int [] array) {
int l = array.length;

int [] temp_array = new int [l];

// Here's the first step.
for (int i=0; i<l-1; i++) {
  temp_array[i]=array[i+1];
}
temp_array[l-1]=array[0];

// Here's the second step.
for (int i=0; i<l; i++) {
  array[i]=temp_array[i];
  }
}
```

Sorting!

- Extremely important application
  - Databases
  - Science
  - Computing medians and percentiles
  - ...
- Seems easy, but how do we make a computer do it?

DrJava