

## CSE 142 Section Handout #7 Solutions

1.	Array	Value Returned
	[5]	0
	[3, 12]	9
	[4, 2, 10, 8]	6
	[1, 9, 3, 5, 7]	8
	[8, 2, 10, 4, 10, 9]	2

2.	Array	Final Contents
	[8]	[8]
	[14, 7]	[14, 8]
	[7, 1, 3, 2, 0, 4]	[7, 2, 3, 3, 1, 4]
	[10, 8, 9, 5, 5]	[10, 9, 9, 6, 6]
	[12, 11, 10, 10, 8, 7]	[12, 12, 11, 11, 9, 8]

3.	Array	Final Contents
	[3, 7, 4]	[3, 3, 4]
	[0, 3, 7, 4, 1]	[0, 3, 3, 2, 1]
	[4, 3, 8, 5, 1, 2]	[4, 3, 4, 2, 1, 2]
	[2, 1, 5, 4, 10, 6, 2]	[2, 1, 2, 4, 5, 6, 2]
	[1, 2, 1, 2, 1, 2, 1]	[1, 1, 1, 1, 1, 1, 1]

#### 4. (two solutions shown)

```
public static int findMin(int[] list) {
    int min = list[0];
    for (int i = 1; i < list.length; i++) {
        if (list[i] < min) {
            min = list[i];
        }
    }
    return min;
}
```

```
public static int findMin(int[] list) {
    int min = list[0];
    for (int i = 1; i < list.length; i++) {
        min = Math.min(min, list[i]);
    }
    return min;
}
```

#### 5.

```
public static boolean isSorted(double[] list) {
    for (int i = 0; i < list.length - 1; i++) {
        if (list[i] > list[i + 1]) {
            return false;
        }
    }
    return true;
}
```

#### 6.

```
public static boolean isPalindrome(String word) {
    word = word.toLowerCase();
    for (int i = 0; i < word.length() / 2; i++) {
        if (word.charAt(i) != word.charAt(word.length() - 1 - i)) {
            return false;
        }
    }
    return true;
}
```

#### 7.

```
public static int[] countLastDigits(int[] list) {
    int[] count = new int[10];
    for (int i = 0; i < list.length; i++) {
        int digit = list[i] % 10;
        count[digit]++;
    }
    return count;
}
```

*(continued on back page)*

## CSE 142 Section Handout #7 Solutions (continued)

### 8. (two solutions shown)

```
public static int[] vowelCount(String text) {
    int[] counts = new int[5];
    for (int i = 0; i < text.length(); i++) {
        char c = text.charAt(i);
        if (c == 'a') {
            counts[0]++;
        } else if (c == 'e') {
            counts[1]++;
        } else if (c == 'i') {
            counts[2]++;
        } else if (c == 'o') {
            counts[3]++;
        } else if (c == 'u') {
            counts[4]++;
        }
    }
    return counts;
}

public static int[] vowelCount(String text) {
    char[] vowels = {'a', 'e', 'i', 'o', 'u'};
    int[] counts = new int[5];
    for (int i = 0; i < text.length(); i++) {
        for (int j = 0; j < vowels.length; j++) {
            if (text.charAt(i) == vowels[j]) {
                counts[j]++;
            }
        }
    }
    return counts;
}
```

### 9. (two solutions shown)

```
public static int[] sum5(int[] list) {
    int[] result = new int[5];
    for (int i = 0; i < list.length; i++) {
        result[i % 5] += list[i];
    }
    return result;
}

public static int[] sum5(int[] list) {
    int[] result = new int[5];
    for (int i = 0; i < 5; i++) {
        for (int j = i; j < list.length; j += 5) {
            result[i] += list[j];
        }
    }
    return result;
}
```

### 10.

```
public static void rotateRight(int[] list) {
    int last = list[list.length - 1];
    for (int j = list.length - 1; j > 0; j--) {
        list[j] = list[j - 1];
    }
    list[0] = last;
}
```

### 11.

```
public static int[] stretch(int[] list) {
    int[] result = new int[2 * list.length];
    for (int i = 0; i < list.length; i++) {
        result[2 * i] = list[i] / 2 + list[i] % 2;
        result[2 * i + 1] = list[i] / 2;
    }
    return result;
}
```