CSE142 Final Exam Key
Autumn 2019

1. The program produces the following output:

```
1 [0, 6, 20]
0 [0, 6, 20]
2 [0, 3, 40]
1 [0, 3, 40]
```

2.

Original Array	Final Array
[9, 9]	[9, 9]
[2, 4, 6, 0, 1]	[2, 4, 6, -1, 1]
[8, 6, 7, 5, 3, 0, 9]	[8, -1, 7, 2, 3, -9, 9]
[4, 8, 15, 16, 23, 42]	[4, 8, 15, 16, 23, 42]
[5, 4, 3, 2, 1]	[5, 1, 3, 1, 1]

3. The program produces the following output:

```
Man
Canal 1 Man 1
Panama 2 Canal 1

Canal
Plan 1 Canal 2
Canal 2

Canal
Canal 1
Canal 2

PanamaCanal
Canal 1
Panama 2 Canal 1
```

4. Three possible solutions:

```
public static void switchPairs(ArrayList<String> list) {
   for (int i = 0; i < list.size() - 1; i += 2) {
      String temp = list.get(i);
      list.set(i, list.get(i + 1));
      list.set(i + 1, temp);
   }
}

public static void switchPairs(ArrayList<String> list) {
   for (int i = 0; i < list.size() / 2; i ++) {
      String temp = list.get(2*i);
      list.set(2*i, list.get(2*i+1));
      list.set(2*i+1, temp);
   }
}

public static void switchPairs(ArrayList<String> list) {
   for (int i = 0; i < list.size() - 1; i += 2) {
      String temp = list.get(i);
      list.add(i + 2, temp);
      list.remove(i);
   }
}
```

5. One possible solution:

```
public static void switchNames(Scanner input) {
   while (input.hasNextLine()) {
      String line = input.nextLine();
      if (line.contains(",")) {
         int comma = line.indexOf(",");
         String last = line.substring(0, comma);
         String first = line.substring(comma + 2);
         System.out.println(first + " " + last);
      } else {
         System.out.println(line);
      }
   }
}
```

6. One possible solution:

```
public static void filter(Scanner input, int min, int max) {
   while (input.hasNext()) {
      String label = input.next();
      int value = input.nextInt();

      if (value < min) {
         System.out.println(label + " - TOO SMALL");
      } else if (value > max) {
         System.out.println(label + " - TOO BIG");
      } else {
         System.out.println(label + " - " + value);
      }
   }
}
```

7. One possible solution:

```
public static boolean isFibLike(int[] list) {
    for (int i = 2; i < list.length; i++) {
        if (list[i] != list[i - 2] + list[i - 1]) {
            return false;
        }
    }
    return true;
}
```

8. One possible solution:

```
public class CandyCane extends Critter {
   private int moves;

   public Action getMove(CritterInfo info) {
      moves++;
      if (moves % 2 == 0 && info.getFront() == Neighbor.OTHER) {
         return Action.INFECT;
      } else if (info.getFront() == Neighbor.EMPTY) {
         return Action.HOP;
      } else if (info.getFront() == Neighbor.SAME) {
         return Action.LEFT;
      } else {
         return Action.RIGHT;
      }
   }
```

```
  public Color getColor() {
     if (moves % 2 == 0) {
        return Color.WHITE;
     } else {
        return Color.RED;
     }
  }

  public String toString() {
     if (moves % 2 == 0) {
        return "J";
     } else {
        return "7";
     }
  }
}
```

9. Three possible solutions:

```
public static int[] delta(int[] a) {
    int[] result = new int[a.length + (a.length - 1)];
    for (int i = 0; i < result.length; i++) {
        if (i % 2 == 0) {
            result[i] = a[i / 2];
        } else {
            result[i] = a[i / 2 + 1] - a[i / 2];
        }
    }
    return result;
}

public static int[] delta(int[] a) {
    int[] b = new int[2 * a.length - 1];
    for (int i = 0; i < a.length; i++) {
        b[2 * i] = a[i];
    }
    for (int i = 1; i < a.length; i++) {
        b[2 * i - 1] = b[2 * i] - b[2 * i - 2];
    }
    return b;
}

public static int[] delta(int[] a) {
    int[] b = new int[a.length + (a.length - 1)];
    for (int i = 0; i < a.length - 1; i++) {
        b[2 * i] = a[i];
        b[2 * i + 1] = a[i + 1] - a[i];
    }
    b[b.length - 1] = a[a.length - 1];
    return b;
}
```

10. Three possible solutions:

```
public static int numWords(String s) {
    int count = 0;
    boolean inWord = false;
    for (int i = 0; i < s.length(); i++) {
        if (s.charAt(i) == ' ')
            inWord = false;
        else if (!inWord) {
            count++;
            inWord = true;
        }
    }
    return count;
}

public static int numWords(String s) {
    int count = 0;
    char prev = ' ';
    for (int i = 0; i < s.length(); i++) {
        if (prev == ' ' && s.charAt(i) != ' ') {
            count++;
        }
        prev = s.charAt(i);
    }
    return count;
}

public static int numWords(String str) {
   int count = 0;
   while (str.length() > 0) {
      int index = str.indexOf(" ");
      if (index != 0) {
         count++;
         if (index != -1) {
            str = str.substring(index);
         } else {
            str = "";
         }
      } else {
         str = str.substring(1);
      }
   }
   return count;
}
```