COS126 Array Activity - Section 1.4

- Complete the program `HowMany.java` (Web Exercise 1.4.1)

```
 1 /*************************************************************************
 2  *  Compilation:  javac HowMany.java
 3  *  Execution:    java HowMany str1 str2 ... strN
 4  *
 5  *  HowMany takes a variable number of command-line arguments
 6  *  and prints a message reporting how many there are.
 7  *
 8  *   > java HowMany
 9  *   You entered 0 command-line arguments.
10  *
11  *   > java HowMany Alice Bob Carol
12  *   You entered 3 command-line arguments.
13  *
14  *   > java HowMany Alice
15  *   You entered 1 command-line argument.
16  *
17  *************************************************************************/
18
19 public class HowMany {
20
21    public static void main(String[] args) {
22
23       // number of command-line arguments
24       int N = ___________________________________;
25
26       // output message
27       System.out.print("You entered " + N + " command-line argument");
28       if (__________) System.out.println(".");
29       else            System.out.println("s.");
30    }
31 }
```

- Complete the program `Distinct.java` below, so that it can take any number of integers as command-line inputs, and check if they are all distinct.

```
 1  /*************************************************************************
 2   * Compile: javac Distinct.java
 3   * Execute: java Distinct int0 int1 int2 ...
 4   *
 5   * Input:   a list of integers
 6   * Output:  true if the inputs all have different values, false otherwise
 7   *
 8   * > java Distinct 11 23 -7 0 99 5 42
 9   * true
10   *
11   * > java Distinct 2 4 6 3 6
12   * false
13   *
14   * > java Distinct -3 -2 -1 -0 3 2 1 0
15   * false
16   * (Note: the reason is that the integer -0 equals the integer 0.)
17   *************************************************************************/
18
19 public class Distinct {
20     public static void main(String[] args) {
21
22         int N = args.length;
23
24         // convert each arg and store them in an array of integers
25         int[] values = _________________________________;
26         for (int i = 0; _______________; _________)
27             ________________ = Integer.parseInt(args[i]);
28
29         // are all of the pairs examined so far distinct?
30         boolean result = true;
31
32         // we'll examine each values[i] in the array
33         for (int i = 0; i < N; i++) {
34             // we'll examine values[j] for each other j
35             for (int j = _____; _________; _________) {
36                 // are they different or not?
37                 if (_________________________________) {
38                     result = _______________;
39                 }
40             }
41         }
42
43         System.out.println(result);
44     }
45 }
```

- **Tracing.** What does this program do? (Stumped? See Exer. 1.4.4.)

```
 1 public class MysteryArray {
 2    public static void main(String[] args) {
 3       int N = args.length;
 4       int[] a = new int[N];
 5
 6       // store the arguments in an integer array
 7       for (int i = 0; i < N; i++) {
 8           a[i] = Integer.parseInt(args[i]);
 9       }
10
11       // What is happening here?
12       for (int i = 0; i < N/2; i++) {
13           int temp = a[i];
14           a[i] = a[N - i - 1];
15           a[N - i - 1] = temp;
16       }
17
18       // print out the elements
19       for (int i = 0; i < N; i++) {
20           System.out.print(a[i] + " ");
21       }
22       System.out.println();
23    }
24 }
```

Write the output for `java MysteryArray 1 3 5 7 9`.

- Complete the program `Birthday.java` (Booksite Creative Exercise 1.4.35)

```
 1: /*************************************************************************
 2:  *  Compilation:  javac Birthday.java
 3:  *  Execution:    java Birthday D
 4:  *
 5:  *  Reads an integer command-line argument D and simulates the number
 6:  *  of people with random birthdays (among D days) that enter a room
 7:  *  until two share a common birthday.
 8:  *
 9:  *   > java Birthday 365
10:  *   22
11:  *************************************************************************/
12:
13: public class Birthday {
14:     public static void main(String[] args) {
15:         // number of days
16:         int D = ________________________________________ ;
17:
18:         // number of people who have entered the room
19:         int people = 0;
20:
21:         // days[d] = true if a person has birthday d; false otherwise
22:         // auto-initialized to false
23:         _______________[] days  =   new  ______________________ ;
24:
25:         // repeat until two people have the same birthday
26:         while (true) {
27:             // increment number of people
28:             people _________ ;
29:
30:             // random day between 0 and D-1
31:             int d = ____________________________________________ ;
32:
33:             // if another person shares birthday d, break out of loop
34:             if ( ___________________________ )   ___________________ ;
35:
36:             // update days[] to indicate person has birthday d
37:             days[ _____ ] = ________________ ;
38:         }
39:
40:         // print result - How many people entered room to get duplication?
41:         System.out.println( ______________________________ );
42:     }
43: }
```

- Additional recommended exercises: `DiscreteDistribution` (template on Precepts page — useful later in the course), `Birthdays` extension of `Birthday` (Booksite Creative Exercise 1.4.35), 1.4.4, 1.4.9, 1.4.10 (Hint: start with Deck.java from lecture), 1.4.13