• Recommended exercises: 1.4.4, 1.4.9, 1.4.10 (Hint: start with Deck.java from lecture), 1.4.13

• Complete the program HowMany.java (Web Exercise 1.4.1)

```java
public class HowMany {
    public static void main(String[] args) {
        // number of command-line arguments
        int N = __________________________________;

        // output message
        System.out.print("You entered " + N + " command-line argument");
        if (__________) System.out.println(".");
        else System.out.println("s.");
    }
}
```

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
Complete the program so it can take an unspecified number of command-line inputs, store them in an integer array, and compute their sum. The program will then output a random index calculated using the inputs as frequency counts. (Web Exercise 1.4.2)

```java
import java.util.Random;

public class DiscreteDistribution {
    public static void main(String[] args) {
        // read in N frequencies. store in integer array.
        int N = __________________________;
        ____________[] freq = __________ int[___];
        for (_____________________; __________________; __________) {
            freq[_____] = Integer.parseInt(__________________);
        }

        // compute total count of all frequencies
        int total = ______;
        for (int i = 0; i < N; i++) {
            total += __________________________________;
        }

        // generate random integer with probability proportional to frequency
        int r = (int) (total * Math.random()); // integer in [0, total)
        int sum = 0;
        int event = -1;
        for (int i = 0; i < N && sum <= r; i++) {
            sum += freq[i];
            event = i;
        }

        System.out.println(event);
    }
}
```

1 /*******************************************************************
2 * Compile: javac DiscreteDistribution.java
3 * Execute: java DiscreteDistribution freq0 freq1 freq2 . . .
4 * Reads in an array of N frequency counts from command line.
5 * Prints out i with probability proportional to ith frequency count.
6 *
7 * // six events, one is 3x more likely than the others
8 * % java DiscreteDistribution 1 1 1 1 1 3 // answer will vary
9 * 5
10 . . .
11 *******************************************************************/
12
13 public class DiscreteDistribution {
14     public static void main(String[] args) {
15         // read in N frequencies. store in integer array.
16         int N = __________________________;
17         ____________[] freq = __________ int[___];
18         for (_____________________; __________________; __________) {
19             freq[_____] = Integer.parseInt(__________________);
20         }
21
22         // compute total count of all frequencies
23         int total = ______;
24         for (int i = 0; i < N; i++) {
25             total += __________________________________;
26         }
27
28         // generate random integer with probability proportional to frequency
29         int r = (int) (total * Math.random()); // integer in [0, total)
30         int sum = 0;
31         int event = -1;
32         for (int i = 0; i < N && sum <= r; i++) {
33             sum += freq[i];
34             event = i;
35         }
36
37         System.out.println(event);
38     }
39 }
40```
• Tracing. What does this program do? (Stumped? See Exer. 1.4.4.)

```java
public class MysteryArray {
    public static void main(String[] args) {
        int N = args.length;
        int[] a = new int[N];

        // store the arguments in an integer array
        for (int i = 0; i < N; i++) {
            a[i] = Integer.parseInt(args[i]);
        }

        // What is happening here?
        for (int i = 0; i < N/2; i++) {
            int temp = a[i];
            a[i] = a[N - i - 1];
            a[N - i - 1] = temp;
        }

        // print out the elements
        for (int i = 0; i < N; i++) {
            System.out.print(a[i] + " ");
        }
        System.out.println();
    }
}
```

Write the output for `java MysteryArray 1 3 5 7 9`. 
Complete the program `Birthday.java` (Booksites Creative Exercise 1.4.35)

```java
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:     }
```

Write a program `Birthdays.java` that takes two integer command-line arguments D and T and repeats the birthday experiment T times, and prints out the average number of people with random birthdays (among D days) that enter a room until two share a common birthday. (Booksities Creative Exercise 1.4.35)