

COS126 Array Activity - Section 1.4

- 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)

```
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 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)

```

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
. . .
26 *****/
27
28 public class DiscreteDistribution {
29     public static void main(String[] args) {
30
31         // read in N frequencies. store in integer array.
32         int N = _____;
33         _____[] freq = _____ int[____];
34         for (_____ ; _____ ; _____) {
35             freq[_____] = Integer.parseInt(_____);
36         }
37
38         // compute total count of all frequencies
39         int total = _____;
40         for (int i = 0; i < N; i++) {
41             total += _____;
42         }
43
44         // generate random integer with probability proportional to frequency
45         int r = (int) (total * Math.random()); // integer in [0, total)
46         int sum = 0;
47         int event = -1;
48         for (int i = 0; i < N && sum <= r; i++) {
49             sum += freq[i];
50             event = i;
51         }
52
53         System.out.println(event);
54     }
55 }

```

- **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: }

```

- 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. (Booksite Creative Exercise 1.4.35)