

CSSE 230 In-class exercise Day 1 Names _____

Work with a partner. If an odd number of students are present, one group may have three people.

1. This question deals with the overhead of having a "growable" array in the **ReadStrings** program (Weiss, pp. 42-43, also printed on the back of this page). Let N be the number of Strings that are actually input when the program is run (N is called *itemsRead* in the program).

Which statement from the code is responsible for most of the "overhead" time associated with growing the array when it is too small to hold the next String?? [Note that all of the array growth originates from calls to `resize()` in line 27. Show the line number of that statement here. If you are not sure that you have the correct line, check with someone else before going on to the rest of part 1.

Line number _____

Let E_N be the number of times (altogether, during the all of the executions of the *while* loop in `getStrings`) that your selected statement is executed for a given value of N . Fill in the middle column of the following table:

N	E_N	Answers for problem 2
4	0	0
5	0	0
6	5	5
7		
10		35
11	15	
20		
21		
40		
41		

In the worst case (for general N), find a formula for E_N as a function of N . This worst case happens when $N=5 \cdot 2^k + 1$. Thus you should only calculate the value of E_N for this case. First write E_N as a summation involving k . Simplify the sum, and then express it in terms of N . The expression should be a simple one

2. Suppose that in the line 27 call to `resize()`, we replace **length * 2** by **length + 1**. In the right column of the table, fill in the numbers that would be the answers to the same question (how many times the "largest overhead" statement executes) in this case. You only need to do this for the first 6 rows of the table (up through $N=11$).
3. Write down a formula (a summation in terms of N) for the number of times the same "maximum execution" statement from part 1 is executed as a function of N in the $\text{length} \leftarrow \text{length} + 1$ case. Again simplify to get a simple closed-form expression involving N .
4. Which would you recommend: the "doubling" or the "add one" approach? Explain.

```

1 import java.io.InputStreamReader;
2 import java.io.BufferedReader;
3 import java.io.IOException;
4
5 public class ReadStrings
6 {
7     // Read an unlimited number of String; return a String [ ]
8     // The minimal I/O details used here are not important for
9     // this example and are discussed in Section 2.6.
10    public static String [ ] getStrings ( )
11    {
12        BufferedReader in = new BufferedReader( new
13            InputStreamReader( System.in ) );
14        String [ ] array = new String[ 5 ];
15        String oneLine;
16        int itemsRead = 0;
17
18        System.out.println( "Enter strings, one per line; " );
19        System.out.println( "Terminate with empty line: " );
20
21        try
22        {
23            while( ( oneLine = in.readLine( ) ) != null &&
24                !oneLine.equals( "" ) )
25            {
26                if( itemsRead == array.length )
27                    array = resize( array, array.length * 2 );
28                array[ itemsRead++ ] = oneLine;
29            }
30        }
31        catch( IOException e )
32        {
33            System.out.println( "Early abort of read." );
34        }
35
36        return resize( array, itemsRead );
37    }
38
39    // Resize a String[ ] array; return new array
40    public static String [ ] resize( String [ ] array,
41        int newSize )
42    {
43        String [ ] original = array;
44        int numToCopy = Math.min( original.length, newSize );
45
46        array = new String[ newSize ];
47        for( int i = 0; i < numToCopy; i++ )
48            array[ i ] = original[ i ];
49        return array;
50    }
51
52    public static void main( String [ ] args )
53    {
54        String [ ] array = getStrings ( );
55        for( int i = 0; i < array.length; i++ )
56            System.out.println( array[ i ] );
57    }

```

Figure 2.7 Code to read an unlimited number of Strings and output them