1. s = ________________

2. ________________

3. ________________

4. ________________

5. ________________

6. □ □ □ □

7. (a) ________________
   (b) ________________

8. (a) ________________
   (b) ________________
   (c) ________________
   (d) ________________

9. □ □ □ □

10. (a) ________________
    (b) ________________
    (c) ________________

11. ________________

12. ________________
This is a closed-book, no-calculator, no-electronic-devices, individual-effort exam. You may reference one page of handwritten notes. All answers should be clearly written. Questions that require code should be written using correct Java syntax. You may write `System.out.println`.

<table>
<thead>
<tr>
<th>Class</th>
<th>Method/Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scanner</td>
<td>Scanner(System.in)</td>
<td>create Scanner for parsing System.in</td>
</tr>
<tr>
<td></td>
<td>Scanner(String text)</td>
<td>create Scanner for parsing text</td>
</tr>
<tr>
<td></td>
<td>String next()</td>
<td>get next delimited word</td>
</tr>
<tr>
<td></td>
<td>double nextDouble()</td>
<td>get next delimited double</td>
</tr>
<tr>
<td></td>
<td>boolean nextBoolean()</td>
<td>get next delimited boolean</td>
</tr>
<tr>
<td></td>
<td>int nextInt()</td>
<td>get next delimited integer</td>
</tr>
<tr>
<td>String</td>
<td>int length()</td>
<td>get number of characters</td>
</tr>
<tr>
<td></td>
<td>char charAt(int i)</td>
<td>get the character at index i</td>
</tr>
<tr>
<td></td>
<td>String toUpperCase()</td>
<td>get a String like this one, but in all-caps</td>
</tr>
<tr>
<td></td>
<td>int indexOf(char c)</td>
<td>find the index of the first c</td>
</tr>
<tr>
<td></td>
<td>String substring(int a)</td>
<td>get substring from index a to String’s end</td>
</tr>
<tr>
<td></td>
<td>String substring(int a, int b)</td>
<td>get substring from index a to before index b</td>
</tr>
<tr>
<td>Math</td>
<td>int max(int a, int b)</td>
<td>get the maximum of a and b</td>
</tr>
<tr>
<td></td>
<td>int min(int a, int b)</td>
<td>get the minimum of a and b</td>
</tr>
<tr>
<td></td>
<td>double pow(double base, double exponent)</td>
<td>raise base to the exponent power</td>
</tr>
<tr>
<td>Random</td>
<td>Random()</td>
<td>create a random number generator</td>
</tr>
<tr>
<td></td>
<td>nextInt(int i)</td>
<td>get random number in [0, i - 1]</td>
</tr>
<tr>
<td></td>
<td>nextDouble()</td>
<td>get random number in [0.0, 1.0)</td>
</tr>
<tr>
<td>Color</td>
<td>Color(int r, int g, int b)</td>
<td>create a color, with each intensity in [0, 255]</td>
</tr>
<tr>
<td>Character</td>
<td>isDigit(char c)</td>
<td>returns true if c is a digit</td>
</tr>
<tr>
<td></td>
<td>isLetter(char c)</td>
<td>returns true if c is uppercase</td>
</tr>
</tbody>
</table>
1. Provide a value for `String s` that makes the following expression true: `s.length() < 6 && s.length() % 2 == 1`.

2. Simplify the expression `!(x > y)`. Be careful.

3. Write a `single` statement that initializes the array `nums` to hold the values 11, 12, and 13.

4. Write a `single` statement that initializes the array `nums` to hold 1000 zeroes.

5. Rewrite the following code to be simpler:

```java
if (isX) {
    if (isY) {
        System.out.println("xy");
    }
}
```

6. Consider the following code:

```java
if (x >= y) {
    System.out.print("x");
} else {
    System.out.print("y");
}
```

Which are possible outputs of this code? Check zero or more.

(a) x  
(b) xy  
(c) y  
(d) no output

7. Fill in the blanks such that a total of 72 asterisks are printed.

```java
for (int r = 0; r < ____a____; ++r) {
    for (int c = 0; c < ____b____; ++c) {
        System.out.print('*');
    }
    System.out.println();
}
```

8. Complete the following truth table. Feel free to write intermediate steps in the blank space.

| x | y | !(x || !y) |
|---|---|-----------|
| 0 | 0 |          |
| 0 | 1 |          |
| 1 | 0 |          |
| 1 | 1 |          |

(a) [Intermediate steps]  
(b) [Intermediate steps]  
(c) [Intermediate steps]  
(d) [Intermediate steps]
9. Consider the following code:

```java
if (x >= y) {
    System.out.print("x");
} else if (y >= x) {
    System.out.print("y");
}
```

Which are possible outputs of this code? Check zero or more.

(a) x  
(b) xy  
(c) y  
(d) no output

10. Fill in the blanks such that the sequence 5, 10, 15, 20, 25 is printed.

```java
for (___a____; ____b____; ___c___) {
    System.out.println(i);
}
```

11. Rewrite the following code to be simpler:

```java
if (isX) {
    System.out.println("xy");
} else if (isY) {
    System.out.println("xy");
}
```

12. Rewrite this for loop as a while loop:

```java
for (int n = 1; n <= 1024; n *= 2) {
    System.out.println(n);
}
```

13. Write a method `chronotype` that accepts an `int` as its only parameter. Return as a `String` the `chronotype` of the person who wakes at the given hour. A "lark" wakes up at 6 o’clock or earlier. An "owl" wakes up at 10 o’clock or later. For everyone else, return "?".

14. Write a method `abc123` that accepts a `String` as its only parameter. It returns a `String` that contains only the alphanumeric characters of the parameter `String`. For example, `abc123("Gods kalpot Latvijai!!!") → "GodskalpotLatvijai"`. Do not use a regular expression.

15. Write a method `pairSum` that accepts an array of `ints`. It returns an array of the pairwise sums of the given array. For example, `pairSum(new int[]{1, 3, 4, 9}) → {4, 13}`, and `pairSum(new int[]{10, 11, 30, 40, 100, 200}) → {21, 70, 300}`. Assume that the given array has an even length.