

## Unit 9. Sorting, Searching, and Recursion Exercises.

1. If you search for the value 30 using a linear search, which indices of the array will you have to check? _____ 2. If you search for the value -18 using a binary search, which indices of the array will you have to check? _____ 3. If you search for the value 90 using a binary search, which indices of the array will you have to check? _____	<b>Index</b>	<b>Value</b>
	0	16
	1	30
	2	88
	3	89
	4	90
	5	130

4. If you search for the value -55 using a binary search, which indices of the array will you have to check? _____ 5. If you search for the value 15 using a binary search, which indices of the array will you have to check? _____ 6. If you search for the value 2 using a linear search, which indices of the array will you have to check? _____ 7. If you search for the value -122 using a linear search, which indices of the array will you have to check? _____	<b>Index</b>	<b>Value</b>
	0	-80
	1	-55
	2	-5
	3	0
	4	2
	5	3
	6	4
	7	12
	8	13

8. Is a zero, positive, or negative number displayed?	<code>System.out.println( "cow".compareTo( "horse" ) );</code>
9. Is a zero, positive, or negative number displayed?	<code>System.out.println( "zoo".compareTo("zebra" ) );</code>
10. Is a zero, positive, or negative number displayed?	<code>System.out.println( "LAX".compareTo( "PHL" ) );</code>
11. Is a zero, positive, or negative number displayed?	<code>System.out.println( "PHL".compareTo( "LAX" ) );</code>
12. Is a zero, positive, or negative number displayed?	<code>System.out.println( "b".compareTo( "boo" ) );</code>

Note. I will not test you on whether a lower case letter is greater than the corresponding upper case letter. I will not test you on whether the character “3” is greater or less than some other string.

```

Student x = new Student("Andy", 123 );
Student y = new Student("Boris", 56 );
if ( x.compareTo( y ) > 0 )
    System.out.println( x + " is greater" );
else if ( y.compareTo( x ) > 0 )
    System.out.println( y + " is greater" );
else
    System.out.println( "same" );

```

13. What does the above code display?

```

public class Student implements Comparable {
    private String name;
    private int id;

    public Student( String s, int i ){
        name = s;
        id = i;
    }

    public int compareTo( Object x ){
        if ( !(x instanceof Student) )
            throw new IllegalArgumentException(
                "Hey! Only students");

        Student joe = (Student) x;
        return name.compareTo( joe.name );
    }

    public String toString(){
        return name + " id: " + id;
    }
}

```

Note. The if statement in the compareTo method is the proper way to detect and handle possible casting errors. However, this is not a testable topic for us. Sometimes I will not include the if statement (like in the next problem).

```

Box2D b1 = new Box2D( 4, 8 );
Box2D b2 = new Box2D( 3, 10 );

int result = b1.compareTo( b2 );
System.out.println( result );

result = b2.compareTo( b1 );
System.out.println( result );

```

14. What does the above code display?

15. Give an example of a situation where the compareTo method does NOT give a result that is consistent with the equals method.

```

public class Box2D implements Comparable {
    private int width, height;

    public Box2D( int w, int h ){
        width = w;
        height = h;
    }

    public int compareTo( Object x ){
        Box2D b = (Box2D) x;
        if ( width == b.width && height == b.height )
            return 0;

        int myarea = width*height;
        int barea = b.width*b.height;
        return myarea - barea;
    }

    public boolean equals( Object x ){
        Box2D b = (Box2D) x;
        return width==b.width && height==b.height;
    }
}

```

16. **Selection Sort.** An array has the values shown in the second column. Complete the table below to show how the contents of the array change (or not) with each pass using a Selection Sort.

array indexes	values in the original order	1 <sup>st</sup> iteration	2 <sup>nd</sup> iteration	3 <sup>rd</sup> iteration	4 <sup>th</sup> iteration
0	8				
1	7				
2	6				
3	5				
4	4				

17. **Insertion Sort.** An array has the values shown in the second column. Complete the table below to show how the contents of the array change (or not) with each pass using an Insertion Sort.

array indexes	values in the original order	1 <sup>st</sup> iteration	2 <sup>nd</sup> iteration	3 <sup>rd</sup> iteration	4 <sup>th</sup> iteration
0	8				
1	7				
2	6				
3	5				
4	4				

18. **Selection Sort.** An array has the values shown in the second column. Complete the table below to show how the contents of the array change (or not) with each pass using a Selection Sort.

array indexes	values in the original order	1 <sup>st</sup> iteration	2 <sup>nd</sup> iteration	3 <sup>rd</sup> iteration	4 <sup>th</sup> iteration
0	4				
1	56				
2	18				
3	77				
4	24				

19. **Insertion Sort.** An array has the values shown in the second column. Complete the table below to show how the contents of the array change (or not) with each pass using an Insertion Sort.

array indexes	values in the original order	1 <sup>st</sup> iteration	2 <sup>nd</sup> iteration	3 <sup>rd</sup> iteration	4 <sup>th</sup> iteration
0	4				
1	56				
2	18				
3	77				
4	24				

*If a particular recursive method never terminates, say so.*

<p>20. What is the value of z? _____  int z = fac( 3 );</p> <p>21. What is the value of y? _____  int y = fac( 5 );</p>	<pre>public int fac( int n ){     if ( n &lt;= 1 )         return 1;     else         return n * fac( n - 1 ); }</pre>
<p>22. What does pop( 8, 8 ) display?</p> <p>23. What does pop( 13, 13 ) display?</p>	<pre>public void pop( int n, int f ) {     if ( n &lt;= 0    f &lt;= 0 )         return;      if ( n % f == 0 )         System.out.print( f + ", " );      pop( n, f - 1 ); }</pre>
<p>24. What is the value of z? _____  int z = abc( 3 );</p> <p>25. What is the value of y? _____  int y = abc( 6 );</p>	<pre>public int abc( int w ) {     if ( w &lt; 0 )         return 0;      return w - abc( w - 2 ); }</pre>
<p>26. What is the value of a? _____  int a = rec( -4 );</p> <p>27. What is the value of b? _____  int b = rec( 2 );</p>	<pre>public int rec( int x ){     if ( x &lt;= 0 )         return x;      else         return x + rec( x + 1 ); }</pre>

<p>28. What is displayed?</p>	<pre>public static void main(String[] args) {     for ( int k = 1; k &lt;= 8; k++ )         System.out.print( fred(k) + " " ); }  public static int fred( int n ) {     if ( n &lt;= 2 )         return 1;     else         return fib( n- 1 ) + fib( n-2 ); }</pre>
<p>29. What does ex1( 78 ) display?</p> <p>30. What does ex1( 1234 ) display?</p>	<pre>public void ex1( int x ){     System.out.print( x%10 + " " );      if ( x/10 != 0 )         ex1( x/10 );      System.out.print( x%10 + " " ); }</pre>
<p>31. What is the value of a? _____              int a = g(8, 16)</p> <p>32. What is the value of b? _____              int b = g(12, 9)</p> <p>33. What is the significance of the number that this method ultimately returns?</p>	<pre>public int g( int x, int y ){     if ( x%y == 0 )         return y;     else         return g( y, x%y ); }</pre> <p>/* This method implements an algorithm from Euclid from about 200 B.C. */</p>
<p>34. What does this display?</p>	<pre>public static void main(String[] args) {     int [] list1 = { 4, 8, 1 };     System.out.println( a1(list1) );      int [] list2 = { 6, -2, 5, 3 };     System.out.println( a1(list2) ); }  public static int a1( int [] a ){     return a1( a, 0 ); }  private static int a1( int [] a, int i ){     if ( i &gt;= a.length )         return 0;     else         return a[i] + a1( a, i+1 ); }</pre>

<p>35. What is displayed?</p> <p>36. Describe, in general terms, what the first m2 method returns.</p>	<pre> public static void main(String[] args) {     int [] a = { 2, 24, 37, 38, 44, 71 };      int x = m2( 17, a );     System.out.println( x );      x = m2( 44, a );     System.out.println( x ); }  public static int m2( int val, int [] a ){     return m2( val, a, 0, a.length-1 ); }  private static int m2( int val, int [] a, int n1, int n2 ) {     if ( n1 &gt; n2 )         return -1;      int num = (n1 + n2)/2;      if ( a[ num ] == val )         return num;     else if ( a[ num ] &lt; val )         return m2( val, a, num + 1, n2 );     else         return m2( val, a, n1, num - 1 ); } </pre>
<p>37 What is displayed?</p> <p>38 Describe, in general terms, what the first mm method returns.</p>	<pre> public static void main(String[] args) {     int [] a = { 3, 7, 2, 6 };      int x = mm( a );     System.out.println( x );      int [] b = { 40, 80, -3 };     x = mm( b );     System.out.println( x ); }  public static int mm( int [] a ){     return mm( a, 0 ); }  private static int mm( int [] a, int n ){     if ( n == a.length-1 )         return a[a.length-1];      int num = mm(a, n+1 );     if ( num &gt; a[n] )         return num;     else         return a[n]; } </pre>