

# UIL COMPUTER SCIENCE WRITTEN TEST

# 2023 INVITATIONAL B

FEBRUARY/MARCH 2023

## General Directions (Please read carefully!)

---

1. DO NOT OPEN THE EXAM UNTIL TOLD TO DO SO.
2. There are 40 questions on this contest exam. You will have 45 minutes to complete this contest.
3. All answers must be legibly written on the answer sheet provided. Indicate your answers in the appropriate blanks provided on the answer sheet. Clean erasures are necessary for accurate grading.
4. You may write on the test packet or any additional scratch paper provided by the contest director, but NOT on the answer sheet, which is reserved for answers only.
5. All questions have ONE and only ONE correct answer. There is a 2-point penalty for all incorrect answers.
6. Tests may not be turned in until 45 minutes have elapsed. If you finish the test before the end of the allotted time, remain at your seat and retain your test until told to do otherwise. You may use this time to check your answers.
7. If you are in the process of actually writing an answer when the signal to stop is given, you may finish writing that answer.
8. All provided code segments are intended to be syntactically correct, unless otherwise stated. You may also assume that any undefined variables are defined as used.
9. A reference to many commonly used Java classes is provided with the test, and you may use this reference sheet during the contest. AFTER THE CONTEST BEGINS, you may detach the reference sheet from the test booklet if you wish.
10. Assume that any necessary import statements for standard Java SE packages and classes (e.g., `java.util`, `System`, etc.) are included in any programs or code segments that refer to methods from these classes and packages.
11. NO CALCULATORS of any kind may be used during this contest.

## Scoring

---

1. Correct answers will receive **6 points**.
2. Incorrect answers will lose **2 points**.
3. Unanswered questions will neither receive nor lose any points.
4. In the event of a tie, the student with the highest percentage of attempted questions correct shall win the tie.

# STANDARD CLASSES AND INTERFACES – SUPPLEMENTAL REFERENCE

## package java.lang

```
class Object
    boolean equals(Object anotherObject)
    String toString()
    int hashCode()

interface Comparable<T>
    int compareTo(T anotherObject)
        Returns a value < 0 if this is less than anotherObject.
        Returns a value = 0 if this is equal to anotherObject.
        Returns a value > 0 if this is greater than anotherObject.

class Integer implements Comparable<Integer>
    Integer(int value)
    int intValue()
    boolean equals(Object anotherObject)
    String toString()
    String toString(int i, int radix)
    int compareTo(Integer anotherInteger)
    static int parseInt(String s)

class Double implements Comparable<Double>
    Double(double value)
    double doubleValue()
    boolean equals(Object anotherObject)
    String toString()
    int compareTo(Double anotherDouble)
    static double parseDouble(String s)

class String implements Comparable<String>
    int compareTo(String anotherString)
    boolean equals(Object anotherObject)
    int length()
    String substring(int begin)
        Returns substring(begin, length()).
    String substring(int begin, int end)
        Returns the substring from index begin through index (end - 1).
    int indexOf(String str)
        Returns the index within this string of the first occurrence of str. Returns
        -1 if str is not found.
    int indexOf(String str, int fromIndex)
        Returns the index within this string of the first occurrence of str, starting
        the search at fromIndex. Returns -1 if str is not found.
    int indexOf(int ch)
    int indexOf(int ch, int fromIndex)
    char charAt(int index)
    String toLowerCase()
    String toUpperCase()
    String[] split(String regex)
    boolean matches(String regex)
    String replaceAll(String regex, String str)

class Character
    static boolean isDigit(char ch)
    static boolean isLetter(char ch)
    static boolean isLetterOrDigit(char ch)
    static boolean isLowerCase(char ch)
    static boolean isUpperCase(char ch)
    static char toUpperCase(char ch)
    static char toLowerCase(char ch)

class Math
    static int abs(int a)
    static double abs(double a)
    static double pow(double base, double exponent)
    static double sqrt(double a)
    static double ceil(double a)
    static double floor(double a)
    static double min(double a, double b)
    static double max(double a, double b)
    static int min(int a, int b)
    static int max(int a, int b)
    static long round(double a)
    static double random()
        Returns a double greater than or equal to 0.0 and less than 1.0.
```

## package java.util

```
interface List<E>
class ArrayList<E> implements List<E>
    boolean add(E item)
    int size()
    Iterator<E> iterator()
    ListIterator<E> listIterator()
    E get(int index)
    E set(int index, E item)
    void add(int index, E item)
    E remove(int index)

class LinkedList<E> implements List<E>, Queue<E>
    void addFirst(E item)
    void addLast(E item)
    E getFirst()
    E getLast()
    E removeFirst()
    E removeLast()

class Stack<E>
    boolean isEmpty()
    E peek()
    E pop()
    E push(E item)

interface Queue<E>
class PriorityQueue<E>
    boolean add(E item)
    boolean isEmpty()
    E peek()
    E remove()

interface Set<E>
class HashSet<E> implements Set<E>
class TreeSet<E> implements Set<E>
    boolean add(E item)
    boolean contains(Object item)
    boolean remove(Object item)
    int size()
    Iterator<E> iterator()
    boolean addAll(Collection<? extends E> c)
    boolean removeAll(Collection<?> c)
    boolean retainAll(Collection<?> c)

interface Map<K,V>
class HashMap<K,V> implements Map<K,V>
class TreeMap<K,V> implements Map<K,V>
    Object put(K key, V value)
    V get(Object key)
    boolean containsKey(Object key)
    int size()
    Set<K> keySet()
    Set<Map.Entry<K, V>> entrySet()

interface Iterator<E>
    boolean hasNext()
    E next()
    void remove()

interface ListIterator<E> extends Iterator<E>
    void add(E item)
    void set(E item)

class Scanner
    Scanner(InputStream source)
    Scanner(String str)
    boolean hasNext()
    boolean hasNextInt()
    boolean hasNextDouble()
    String next()
    int nextInt()
    double nextDouble()
    String nextLine()
    Scanner useDelimiter(String regex)
```

# STANDARD CLASSES AND INTERFACES – SUPPLEMENTAL REFERENCE

Package `java.util.function`

**Interface `BiConsumer<T,U>`**  
`void accept(T t, U u)`

**Interface `BiFunction<T,U,R>`**  
`R apply(T t, U u)`

**Interface `BiPredicate<T,U>`**  
`boolean test(T t, U u)`

**Interface `Consumer<T>`**  
`void accept(T t)`

**Interface `Function<T,R>`**  
`R apply(T t)`

**Interface `Predicate<T>`**  
`boolean test(T t)`

**Interface `Supplier<T>`**  
`T get()`

# UIL COMPUTER SCIENCE WRITTEN TEST – 2023 INVITATIONAL B

**Note:** Correct responses are based on **Java SE Development Kit 17 (JDK 17)** from Oracle, Inc. All provided code segments are intended to be syntactically correct, unless otherwise stated (e.g., "error" is an answer choice) and any necessary Java SE 17 Standard Packages have been imported. Ignore any typographical errors and assume any undefined variables are defined as used. For all output statements, assume that the **System** class has been statically imported using: `import static java.lang.System.*;`

## Question 1

What is the sum of  $10010111_2$  and  $10100011_2$ ?

- A)  $1410_{16}$       B)  $DA_{16}$       C)  $13A_{16}$       D)  $19A_{16}$       E)  $1310_{16}$

## Question 2

What is the output of the code segment to the right?

- A) 3      B) 4      C) 5      D) 6      E) 6.25

```
out.print(1 + 29 % 10 / 4 + 3);
```

## Question 3

What is the output of the code segment to the right?

A) Blue  
Red  
GreenYellow  
PurpleOrange

B) Blue  
Red  
Green  
Yellow  
PurpleOrange

C) Blue  
Red  
Green  
YellowPurple  
Orange

D) BlueRed  
GreenYellowPurpleOrange

E) Blue  
Red  
GreenYellowPurple  
Orange

```
out.print("Blue\nRed\nGreen");  
out.print("Yellow\nPurple");  
out.print("Orange");
```

## Question 4

What is the output of the code segment to the right?

- A) 1      B) 2      C) 7      D) 8      E) 9

```
String str = "Scholastic";  
out.print(str.indexOf("c", 2));
```

## Question 5

What is the output of the code segment to the right?

- A) true  
B) false

```
boolean A = true;  
boolean B = !A;  
out.print(!A && (A || B) ^ B);
```

## Question 6

What is the output of the code segment to the right?

- A) 5      B) 5.0      C) 6      D) 6.0      E) 15

```
double M = Math.sqrt(31);  
out.print((int)Math.floor(M));
```

<p><b>Question 7</b></p> <p>What is the output of the code segment to the right?</p> <p><b>A)</b> 6.4   <b>B)</b> 3.2   <b>C)</b> 6.2   <b>D)</b> 6.0   <b>E)</b> 6</p>	<pre>double T = 2.2; double A = 8 * 4 / 10 + 1 + T ; out.print(A);</pre>
<p><b>Question 8</b></p> <p>What is the output of the code segment to the right?</p> <p><b>A)</b> B <b>B)</b> BCDEF <b>C)</b> BCDEFG <b>D)</b> F <b>E)</b> FG</p>	<pre>int Q = 20 % 7; switch (Q) {     case 1:out.print("A");     case 2:out.print("B");     case 3:out.print("C");     case 4:out.print("D");     case 5:out.print("E");     case 6:out.print("F");     default:out.print("G"); }</pre>
<p><b>Question 9</b></p> <p>How many x's will be output of the code segment to the right?</p> <p><b>A)</b> 4 <b>B)</b> 5 <b>C)</b> 6 <b>D)</b> 7 <b>E)</b> 12</p>	<pre>for(int x = 50; x &gt; 2; x = x / 2 + 1)     out.print("x");</pre>
<p><b>Question 10</b></p> <p>What is the output of the code segment to the right?</p> <p><b>A)</b> 49   <b>B)</b> 45   <b>C)</b> 41   <b>D)</b> 40   <b>E)</b> 36</p>	<pre>int[] four = {4,8,12,16,20,24,28}; int[] five = {5,10,15,20,25,30,35}; if (four.length&lt;five.length)     out.print(four[4]+five[5]); else     out.print(four[5]+five[4]);</pre>
<p><b>Question 11</b></p> <p>What is output by the code segment to the right?</p> <p><b>A)</b> 3684 <b>B)</b> 12243648 <b>C)</b> 361224 <b>D)</b> 363648 <b>E)</b> 3636</p>	<pre>String St = "12 24 36 48"; Scanner go = new Scanner(St); out.print(go.nextInt() + go.nextInt()); out.print(go.next() + go.next());</pre>
<p><b>Question 12</b></p> <p>What is the output of the code segment to the right?</p> <p><b>A)</b> 113   <b>B)</b> 123   <b>C)</b> 73   <b>D)</b> 197   <b>E)</b> 63</p>	<pre>int sum = 0; for(int x = 1; x &lt;= 50; x = x * 2 )     sum += x + 10; out.print(sum);</pre>
<p><b>Question 13</b></p> <p>What is the output of the code segment to the right?</p> <p><b>A)</b> 11   <b>B)</b> 12   <b>C)</b> 13   <b>D)</b> 14   <b>E)</b> 15</p>	<pre>int a = 15, b = 12, c = 9 ; out.print(c ^ b   a - c &amp; b ^ a);</pre>
<p><b>Question 14</b></p> <p>What is the output of the code segment shown on the right?</p> <p><b>A)</b> 0   <b>B)</b> 1   <b>C)</b> -1   <b>D)</b> 127   <b>E)</b> -129</p>	<pre>out.println(Byte.MIN_VALUE + 127);</pre>

<p><b>Question 15</b></p> <p>What is output by the code segment to the right?</p> <p>A) [12, 10, 14, 16, 10]  B) [10, 12, 10, 14, 16, 10]  C) [12, 12, 14, 16, 10]  D) [10, 12, 12, 14, 16, 10]  E) [10, 12, 14, 16, 10]</p>	<pre>ArrayList&lt;Integer&gt; messi; messi = new ArrayList&lt;Integer&gt;(); messi.add(10); messi.add(12); messi.add(messi.get(1)); messi.add(14); messi.remove(0); messi.add(16); messi.add(10); messi.add(0, messi.get(0)-2); out.println(messi);</pre>
<p><b>Question 16</b></p> <p>What is the output of the code segment shown on the right?</p> <p>A) GJ    B) IJ    C) JB    D) DE    E) ED</p>	<pre>String D = "ABCDEFGH IJ"; for(int x=1; x&lt;=4; x++)     D = D.substring(3)+D.substring(0,1); out.println(D);</pre>
<p><b>Question 17</b></p> <p>What is the output of the code segment shown on the right?</p> <p>A) 169    B) 144    C) 121    D) 100    E) 81</p>	<pre>ArrayList&lt;Integer&gt;T; T = new ArrayList&lt;Integer&gt;(); T.add(3); T.add(5); for (int x=2;x&lt;=100;x++) {     int N = T.get(x-2) + T.get(x-1);     T.add(N); } out.print(T.get(8));</pre>
<p><b>Question 18</b></p> <p>What is the output of the code segment shown on the right?</p> <p>A) 3    B) 6    C) 9    D) 12    E) 14</p>	<pre>String St = ""; for(char ch = 'A'; ch&lt;='C'; ch++)     St += ch + St + ch; out.print(St.length());</pre>
<p><b>Question 19</b></p> <p>What is the output of the code segment shown on the right?</p> <p>A) 3    B) 4    C) 5    D) 6    E) 7</p>	<pre>int[] G = {5,1,2,9,2,6,7,4,1,7}; for(int x=1; x&lt;=8; x++)     G[x] = (G[x-1] + G[x+1])/2; out.println(G[5]);</pre>

<p><b>Question 20</b></p> <p>In the code segment to the right, what is the output of line 1?</p> <p>A) 0      B) 1      C) 2      D) 3      E) 4</p>	
<p><b>Question 21</b></p> <p>In the code segment to the right, what is the output of line 2?</p> <p>A) 0      B) 1      C) 2      D) 3      E) 4</p>	<pre>int[] cool = {17,19,12,8,3}; int[] list = new int[cool.length]; for(int x=0;x&lt;cool.length;x++)     for(int y=x+1;y&lt;cool.length;y++)         if(cool[x]&gt;=cool[y])             list[x]++;         else             list[y]++; out.print(list[0]); //line 1 out.print(list[4]); //line 2</pre>
<p><b>Question 22</b></p> <p>In the code segment to the right, if the first line:  <code>int[] cool = {17,19,12,8,3}</code>  was changed to  <code>int[] cool = {17,19,12,8,3,1,22,11,5,6}</code>  What would be the final value of <code>list[0]</code>?</p> <p>A) 1  B) 3  C) 5  D) 7  E) 9</p>	
<p><b>Question 23</b></p> <p>What is the output of the code segment shown on the right?</p> <p>A) CAB    B) BAC    C) 6      D) 216    E) 198</p>	<pre>String St = "CAB"; int N = St.charAt(2); N += St.charAt(1); N += St.charAt(0); out.print(N);</pre>
<p><b>Question 24</b></p> <p>In the code on the right, how many *s will be printed?</p> <p>A) 770  B) 28  C) 800  D) 700  E) 880</p>	<pre>for(int A = 1; A &lt;= 10; A++)     for(int B = -5; B &lt;= 5; B++)         for(int C = 8; C &gt; 1; C--)             out.print("*");</pre>
<p><b>Question 25</b></p> <p>What is returned by the method call <code>Yes(5, 2)</code></p> <p>A) 2      B) 5      C) 7      D) 10     E) 6</p>	<pre>public static int Yes(int x, int y) {     if (x&gt;y)         return x * y;     if (x == y)         return Yes(x+y, y+1);     else         return Yes(x+1, y-1) + 2; }</pre>
<p><b>Question 26</b></p> <p>What is returned by the method call <code>Yes(3, 3)</code></p> <p>A) 9      B) 24     C) 6      D) 12     E) 18</p>	
<p><b>Question 27</b></p> <p>What is returned by the method call <code>Yes(0, 7)</code></p> <p>A) 20     B) 24     C) 26     D) 28     E) 30</p>	

<p><b>Question 28</b></p> <p>In the code to the right, what is output on line #1?</p> <p>A) 10      B) 12      C) 14      D) 16      E) 18</p>	<pre> TreeMap&lt;Character,Integer&gt; Cup; Cup = new TreeMap&lt;Character,Integer&gt;(); </pre>
<p><b>Question 29</b></p> <p>In the code to the right, what is output on line #2?</p> <p>A) 10 B) 12 C) 14 D) 16 E) 18</p>	<pre> Stack&lt;Integer&gt; Bowl; Bowl = new Stack&lt;Integer&gt;();  Cup.put('D',18); Cup.put('A',12); Cup.put('C',14); Cup.put('B',10); Cup.put('C',16); </pre>
<p><b>Question 30</b></p> <p>In the code to the right, what is output on line #3?</p> <p>A) [16, 18] B) [18, 12] C) [12, 16] D) [18, 14] E) [18, 16]</p>	<pre> char x; for (x='D';x&gt;='A';x--)     Bowl.push(Cup.get(x));  out.println(Bowl.pop()); // Line 1 Bowl.pop();  out.println(Cup.get('C'));// Line 2  out.println(Bowl); // Line 3 </pre>
<p><b>Question 31</b></p> <p>What is the output of the code segment shown on the right?</p> <p>A) false false B) false true C) true false D) true true E) no output - there is a compile error</p>	<pre> boolean A = false; boolean B = true; for (int x = 1; x&lt;=12; x++)     if(x*x%2==0)         A = B;     else         B = !B;  out.println(A + " " + B); </pre>



**Question 32**

In the code to the right, what is output by line #1?

- A) 5
- B) 6
- C) 12
- D) 14
- E) 16

**Question 33**

In the code to the right, what is output by line #2?

- A) 12
- B) 24
- C) 28
- D) 36
- E) 42

**Question 34**

In the code to the right, what is output by line #3?

- A) 20
- B) 40
- C) 54
- D) 60
- E) 72

```
public class Soccer
{
    private int A;
    private int B;
    private int C;

    public Soccer(int x, int y, int z)
    {
        A = x;
        B = y;
        C = z;
    }

    public Soccer(int x, int y)
    {
        this(y, x, x+y);
    }

    public Soccer(int x)
    {
        this(x, 2*x);
    }

    public Soccer()
    {
        this(10);
    }

    public int primetime()
    {
        return A+B+C;
    }
}

//client code
Soccer R = new Soccer(1,2,3);
out.println(R.primetime()); // #1
Soccer S = new Soccer(4,8);
out.println(S.primetime()); // #2
Soccer T = new Soccer();
out.println(T.primetime()); // #3
```

<p><b>Question 35</b></p> <p>What is the output of the code segment shown on the right?</p> <p>A) 0    B) 1    C) 7    D) 17    E) 64</p>	<pre>int B = 127; B = B &gt;&gt; 5; B = B &lt;&lt; 4; B = B + 8; B = B &gt;&gt; 3; out.println(B);</pre>
<p><b>Question 36</b></p> <p>What is the output of the code segment shown on the right?</p> <p>A) UNI    B) NIV    C) NI    D) SIT    E) SI</p>	<pre>String St = "UNIVERSITY"; int L = St.indexOf("I"); out.print(St.substring(L-1,L+1));</pre>
<p><b>Question 37</b></p> <p>What is the output of the code segment shown on the right?</p> <p>A) 44 66 11 33 22  B) 44 44 44 44 44  C) 22 22 22 22 22  D) 22 33 11 66 44  E) 44 66 11 66 44</p>	<pre>int[] Pogo = {22, 33, 11, 66, 44}; int x = 10; int y = 4; for(x = 0; x&lt;=4; x++) {     Pogo[x]=Pogo[y];     Pogo[y]=Pogo[x];     y--; } for(int i = 0; i&lt;=4; i++)     out.print(Pogo[i]+" ");</pre>
<p><b>Question 38</b></p> <p>What is the output of the code segment shown on the right?</p> <p>A) 5    B) 50    C) 65    D) 500    E) 650</p>	<pre>int A = 50; int B = 65; int F = -1; for (int x=1; x&lt;=A; x++)     if(A%x==0 &amp;&amp; B%x==0)         F = x; out.print(A * B / F);</pre>
<p><b>Question 39</b></p> <p>Evaluate the prefix expression to the right. Write your answer in the answer blank for #39.</p>	<p><b>+ * / - 90 20 10 3 7</b></p>
<p><b>Question 40</b></p> <p>To the right, we have begun to list all the 6-digit binary numbers from 000000 to 111111. So far, we have listed only 8 of them. If we were to list all 64 of these 6-digit binary numbers, how many "ones" would be written? So far, we have written 12 "ones"</p>	<p><b>000000, 000001, 000010, 000011  000100, 000101, 000110, 000111</b></p>

# ★ ANSWER KEY – CONFIDENTIAL ★

## UIL COMPUTER SCIENCE – 2023 INVITATIONAL B

Questions (+6 points for each correct answer, -2 points for each incorrect answer)

1) C	11) D	21) A	31) D
2) D	12) B	22) D	32) B
3) A	13) E	23) E	33) B
4) E	14) C	24) A	34) D
5) B	15) D	25) D	35) C
6) A	16) A	26) B	36) C
7) C	17) B	27) A	37) E
8) E	18) E	28) B	38) E
9) C	19) D	29) D	*39) 28
10) A	20) D	30) E	*40) 192

KEY

\* See "Explanation" section below for alternate, acceptable answers.

**Note:** Correct responses are based on **Java SE Development Kit 17 (JDK 17)** from Oracle, Inc. All provided code segments are intended to be syntactically correct, unless otherwise stated (e.g., "error" is an answer choice) and any necessary Java SE 17 Standard Packages have been imported. Ignore any typographical errors and assume any undefined variables are defined as used.

### Explanations:

1.	C	<p>Convert all to Base 16 and then add.</p> $10010111_2 = 97_{16}$ $10100011_2 = A3_{16}$ $7 + 3 = 10$ which would be an A $9 + A (10) = 19$ which is 13 (3 carry the 1) $97_{16} + A3_{16} = 13A_{16}$
2.	D	<p>Use order of operations.          Perform integer modulus first.  <math>1 + 29 \% 10 / 4 + 3</math>  <math>1 + 9 / 4 + 3</math>          Next do integer division.  <math>1 + 2 + 3</math>          Now add left to right.  <math>1 + 2 + 3 = 6</math></p>
3.	A	<p>Since there are only print statements, new lines will only be invoked at the <code>\n</code> new line escape character. So new lines will occur after Blue, Red, and Yellow</p>
4.	E	<p><code>str.indexOf("c",2)</code>          This will find the position of "c" if you start looking at position 2. Therefore it "misses" the first "c" and finds the "c" in position 9</p>
5.	B	<p>A is true. B is "not A" meaning B is false.          Now.  <code>!A &amp;&amp; (A    B) ^ B</code>          First, substitute all the values  <code>!true &amp;&amp; (true    false) ^ false</code>          Next, evaluate <code>!true</code>  <code>false &amp;&amp; (true    false) ^ false</code>          Now, do the parenthesis  <code>false &amp;&amp; true ^ false</code>          Next is "xor"  <code>false &amp;&amp; true</code>          Using and          false is the final answer</p>
6.	A	<p><code>Math.sqrt(31)</code> returns a value greater than 5.0 but less than 6.0  <code>Math.floor()</code> will round that value down to 5.0  <code>(int)</code> type casts that value as an integer 5</p>
7.	C	<p>Even though A is a double, the evaluation begins with integer operations.  <math>8 * 4</math> is 32, then <math>32 / 10</math> is 3 using integer division.          Next add 1 (another integer) to get 4 before a double is finally introduced giving us 6.2</p>
8.	E	<p><math>20 \% 7</math> is 6          So case "6" is invoked.          Because there are no break statements, the output will begin with F and keep doing all the inputs through the end of the switch statement.</p>
9.	C	<p><math>x = x / 2 + 1</math>          The values of x in the iterations would be:          50 26 14 8 5 3 then the value 2 would stop the process, but not print a 7th x.</p>
10.	A	<p>The two arrays are the same length, so the else is invoked.          Remember the first element in an array has an index of 0.  <code>four[5]</code> is 24 <code>five[4]</code> is 25          The sum is 49</p>

11.	D	<p>St = "12 24 36 48";  The Scanner go accesses the String St.  The first two go.nextInt() calls treat the inputs as integers and thus adds 12 + 24 to get 36  The next two go.next() calls treat the inputs as Strings and thus adds "36"+"48" = "3648"  So, we get 363648 as the output</p>
12.	B	<p>The loop itself runs through the values 1 2 4 8 16 32.  The accumulator adds these values one by one.  That gets a total of 63  But with every iteration, 10 more is added, for a total of 60.  The output is 63 + 60 = 123</p>
13.	E	<p>a = 15, b = 12, c = 9 ;  a=1111 b=1100 c=1010  c ^ b   a - c &amp; b ^ a  9 ^ 12   15 - 9 &amp; 12 ^ 15  9 ^ 12   <b>15 - 9</b> &amp; 12 ^ 15 (first, subtract)  9 ^ 12   6 &amp; 12 ^ 15 (now it is bit-wise time)  1001 ^ 1100   0110 &amp; 1100 ^ 1111  1001 ^ 1100   <b>0110 &amp; 1100</b> ^ 1111 (do and)  1001 ^ 1100   0100 ^ 1111  <b>1001 ^ 1100</b>   0100 ^ 1111 (do leftmost xor)  0101   0100 ^ 1111  0101   <b>0100 ^ 1111</b> (do other xor)  0101   1011 (finish with the or)  1111 = 15</p>
14.	C	<p>Byte values range from -128 to 127  Byte_MIN_VALUE = -128  -128 + 127 = -1</p>
15.	D	<p>Here is the progression of messi.  [ ]  [10]  [10,12]  [10,12,12]  [10,12,12,14]  [12,12,14]  [12,12,14,16]  [12,12,14,16,10]  [10,12,12,14,16,10]</p>
16.	A	<p>In each iteration, the string loses the first three letters, but that first letter is saved as it moves to the end.  There are 4 iterations.  (0) ABCDEFGHIJ  (1) DEFGHIJA  (2) GHIJAD  (3) JADG  (4) GJ</p>
17.	B	<p>This is a Fibonacci algorithm starting with a 3 and a 5.  Each new element is the sum of the previous two elements.  3 5 8 13 21 34 55 89 144 233 377  The code prints element #8</p>
18.	E	<p>The loop iterates 3 times. For A, B, then C.  A: St becomes "AA"  B: St becomes AABAAB  C: St becomes AABAABCAABAABC</p> <p>The length is 14</p>

19.	D	<p>[5,1,2,9,2,6,7,4,1,7] = original array  Loop goes from 1 to 8. Watch the array change.  1 - [5,3,2,9,2,6,7,4,1,7]  2 - [5,3,6,9,2,6,7,4,1,7]  3 - [5,3,6,4,2,6,7,4,1,7]  4 - [5,3,6,4,5,6,7,4,1,7]  5 - [5,3,6,4,5,6,7,4,1,7]  6 - [5,3,6,4,5,6,5,4,1,7]  7 - [5,3,6,4,5,6,5,3,1,7]  8 - [5,3,6,4,5,6,5,3,5,7]  Element #5 is a 6</p>
20.	D	<p>This is a portion of a sort routine called the "flag-tag" or "Supreme Court" sort.  The elements in the list array represent the indices of the where each corresponding cool item should be in a sorted array.  17 is greater than 3 other values in the cool: 12, 8, and 3</p>
21.	A	3 is greater than 0 other values in cool
22.	D	17 is larger than 7 items 12,8,3,1,11,5,6
23.	E	<p>This adds the ASCII codes of 'A', 'B', and 'C'  It is nice to know that the ASCII Code for 'A' is 65 and 'a' is 97.  Here we add 65+66+67 = 198</p>
24.	A	<p>The A loop iterates 10 times.  For each iteration of the A loop, the B loop iterates 11 times.  For each iteration of the B loop, the C loop iterates 7 times.  <math>10 * 11 * 7 = 770</math></p>
25.	D	<p>Yes(5,2) goes straight to the stopping state condition.  <math>5 * 2 = 10</math></p>
26.	B	<p>Yes (3,3) first goes to the middle option.  Yes(3,3) = Yes(6,4)  Yes(6,4) = 24</p>
27.	A	<p>Yes(0,7) = Yes(1,6) + 2  Yes(1,6) = Yes(2,5) + 2  Yes(2,5) = Yes(3,4) + 2  Yes(3,4) = Yes(4,3) + 2  Yes(4,3) = 12  <math>12 + 2 + 2 + 2 + 2 = 20</math></p>
28.	B	<p>After the 5 Cup.put() lines, we have:  'A' = 12 'B'=10 'C'=16 'D' = 18  These are pushed onto the Bowl stack in reverse order with 18 being on the bottom.  At this point the stack is [18,16,10,12]  The first pop removes the 12</p>
29.	D	<p>Continuing from #28  Cup.get('C') gives us 16</p>
30.	E	<p>Continuing from #28  Cup is now [18,16,10]  10 is popped.  Cup is now [18,16]</p>

31.	D	<p>The loop will test the numbers 1,4,9,16,25,36,49,64,81,100,121,144  Notice that the alternate from odd to even  A = false, B=true.  if even, A takes on B's value  if odd, B's value flips  1 odd - now B=false  4 even - now A=false  9 odd - now B=true  16 even - now A=true  25 odd - now B=false  36 even - now A=false  49 odd - now B=true  64 even - now A=true  81 odd - now B=false  100 even - now A=false  121 odd - now B=true  144 even - now A=true</p>
32.	B	<p>R utilizes the 3-parameter constructor sending in 1,2,3  A=1 B=2 C=3 primetime returns 1+2+3 = 6</p>
33.	B	<p>S utilizes the 2-parameter constructor sending in 4,8  It then calls the 3-parameter constructor sending in 8,4,12  A=8 B=4 C=12 primetime returns 8+4+12 = 24</p>
34.	D	<p>T utilizes the 0-parameter constructor  It then calls the 1-parameter constructor sending in 10  Then it calls the 2-parameter constructor sending in 10,20  It then calls the 3-parameter constructor sending in 20,10,30  A=20 B=10 C=30 primetime returns 20+10+30 = 60</p>
35.	C	<p>B = 127 (1111111)  Each right shift &gt;&gt;divides by 2 using integer division  Each left shift &lt;&lt; multiplies by 2  B &gt;&gt; 5 (right shift 5) divides 127/32 = 3 (11)  B &lt;&lt; 4 (left shift 4) multiples 3 * 16 = 48 (110000)  B + 8 Now add 8 48+8 = 56 (111000)  B &gt;&gt;3 (right shift 3) divides by 8 56/8 = 7 (111)</p>
36.	C	<p>St = "UNIVERSITY"  L will have a value of 2, the location of the first "I"  St.substring(1,3) is "NI"</p>
37.	E	<p>The original values are [22,33,11,66,44]  The loop iterates 5 times.  Pogo[0] becomes Pogo[4] [44,33,11,66,44]  The next line has no effect Pogo[4] = Pogo[0] since they are already equal.  The second iteration makes Pogo[1] = Pogo[3] [22,66,11,66,44]  Again the next line has no effect.  In fact, no other changes are made because:  Pogo[2] = Pogo[2] causes no change.  Pogo[3] = Pogo[1] causes no change.  Pogo[4] = Pogo[0] causes no change.</p>
38.	E	<p>The loop is designed to find the greatest common factor of 50 and 65  Thus F = 5  Now doing 50 * 65 / 5 you get 650  This is a good way to get the least common multiple since  A * B = GCF(A,B) * LCM(A,B)</p>

39.	28	<p>+ * / - <b>90 20</b> 10 3 7</p> <p>+ * / <b>70</b> 10 3 7</p> <p>+ * / <b>70 10</b> 3 7</p> <p>+ * <b>7 3</b> 7</p> <p>+ <b>21 7</b></p> <p><b>28</b></p>
40.	192	<p>If we listed all 64 binary numbers from 000000 to 111111, we would be listing 384 digits (64*6).  If it is a complete list, there would be half zeros and half ones.  Thus, we would have 192 of each.</p>