UIL COMPUTER SCIENCE WRITTEN TEST

2022 REGION

APRIL 2022

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 – 2022 REGION

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.

Which of the following is equal to $101101_2 * 10010_2$?

A) 11010001010_2 **B)** 1100101010_2 **C)** 11010101010_2 **D)** 10010001010_2 **E)** None of these.

Question 2.

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

A) `50` **B)** `50.5` **C)** `10` **D)** `108`

E) There is no output due to a syntax error

```
out.println(17 % 12 * 23 - 14 / 2);
```

Question 3.

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

A) `%%%%2.72`
B) `%%2.72`
C) `2.72`
D) `%%%%2.7`

E) There is no output due to a compile error

```
out.printf("%%%%%.2f", 2.718);
```

Question 4.

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

A) `true` **B)** `false` **C)** `5`

D) There is no output due to a compile error

E) There is no output due to a runtime error

```
out.print("grapple".endsWith("apple", 5));
```

Question 5.

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

A) `true`

B) `false`

```
boolean a = true;
boolean b = false;
out.print(!(a || b) ^ (!a && !b));
```

Question 6.

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

A) `7.0` **B)** `8.0` **C)** `9.0` **D)** `27.0`

E) There is no output due to a compile error

```
out.print(Math.cbrt(27*27));
```

Question 7.

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

A) `1` **B)** `1.0` **C)** `1.5` **D)** `2`

E) There is no output due to a compile error

```
int x = 5;
byte y = 2;
y += 2.5;
out.print(x - y);
```

Question 8.

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

A) 1

B) 2

C) 3

D) 6

E) There is no output due to a compile error

```
int r = 1;
switch("case")
{
  case "c": r++;
  case "ca": r++;
  case "cas": r++;
  case "case": r++;
  default: r++;
}
out.print(r);
```

Question 9.

How many asterisks are printed by the code shown to the right?

A) 20 **B)** 21 **C)** 40 **D)** 42 **E)** 166

```
for(int x = 7; x <= 89; x += 4)
{
  out.print("**");
}
```

Question 10.

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

A) [S, I, L, C, L]

B) [C, I, S, U, U]

C) [C, I, L, L, U]

D) [C, I, L, L, S]

E) None of the above

```
char[] chars = "UILCS".toCharArray();
chars[0] = chars[4];
chars[4] = chars[2];
Arrays.sort(chars);
out.print(Arrays.toString(chars));
```

```
public static void main(String[] args) <CODE>
{
      Scanner f = new Scanner(new File("data.dat"));
}
```

Question 11.

Consider the main method shown above. Which of the following could replace the section marked **<CODE>** to compile without error?

I) catch FileNotFoundException
II) throws FileNotFoundException
III) throws IOException
IV) throws Exception
V) No additional code is needed

A) I

B) II

C) V

D) II and IV

E) II, III, and IV

Question 12.

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

A) 127

B) 128

C) 511

D) 512

E) None of the above

```
int p = 1;
int s = 0;
for(int x = 0;x<=8;x++)
{
  s += p;
  p *= 2;
}
out.print(s);
```

Question 13.

What is the correct order of operations for the operators listed on the right?

A) I II III

B) II III I

C) III II I

D) I III II

E) III I II

I. ~ unary complement

II. << left shift

III. + addition

Question 14.

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

A) -128

B) -127

C) 0

D) 1

E) 127

```
out.print(~Byte.MAX_VALUE);
```

Question 15.

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

A) 12

B) 16

C) 17

D) 21

E) None of the above

```
ArrayList<Integer> list;
list = new ArrayList<>();
for(int i = 0;i<6;i++){
  list.add(10*i + 1);
  list.add(10*i + 4);
  list.add(10*i + 5);
  list.add(10*i + 6);
  list.add(10*i + 9);
}
out.println(list.get(15) - list.get(7));
```

Question 16.

Which of the following is the *best estimate* for the output of the code to the right?

A) 0.125

B) 0.25

C) 0.5

D) 0.75

E) 1.0

```
Random r = new Random();
double ct = 0;
double it = 10000;
for(int i=1;i<=it;i++)
{
  double x = r.nextDouble();
  double y = r.nextDouble();
  if(y < .5 & x < y)
    ct++;
}
out.print(ct / it);
```

Question 17.

Which of the following is the *best estimate* of the output of the code segment shown on the right?

A) true

B) false

C) 5

D) There is no output due to a compile error

E) There is no output due to a runtime error

```
out.print(5 instanceof Integer);
```

Question 18.

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

A) -1

B) 4

C) 5

D) 6

E) 7

```
int[] ints = new int[] {1,4,6,8,9,11,15};
int s = 10;
int L = -1;
int R = ints.length;
int M = (L + R) / 2;
while(R - L > 1)
{
  /*LINE:*/
  M = (L + R) / 2;
  int c = ints[M];
  if(c <= s)
    L = M;
  else
    R = M;
}
out.print(L);
```

Question 19.

How many times does the line labeled **/*LINE:*/** execute before terminating in the code to the right?

A) 0

B) 1

C) 2

D) 3

E) 4

Question 20.

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

A) true 3

B) false 3

C) true 5

D) false 5

E) There is no output due to a compile error

```
TreeSet<Integer> set1;
set1 = new TreeSet<Integer>();
HashSet<Integer> set2;
set2 = new HashSet<Integer>();
for(int i = 0;i<5;i++)
{
  set1.add(i % 3);
  set2.add((i + 2) % 3);
}
boolean eq = set1.equals(set2);
out.print(eq+" "+set2.size());
```

Question 21.

What is the output of the client code below?

```
int[] ints = new int[]{1,4,1,5,1,1,7};
out.print(dp(ints));
```

A) 1

B) 3

C) 4

D) 5

E) 7

Question 22.

What is the output of the client code below?

```
int[] ints = new int[]{3,1,4,1,5,9};
out.print(rec(ints, 0));
```

A) 1

B) 3

C) 4

D) 5

E) 6

Question 23.

Which of the following is the most restrictive Big O upper bound of method call `dp(ints)`? Assume `ints` is of length `N`

A) $O(N)$

B) $O(N\log N)$

C) $O(N^2)$

D) $O(N^3)$

E) $O(2^N)$

Question 24.

Which of the following is the most restrictive Big O upper bound of method call `rec(ints)`? Assume `ints` is of length `N`

A) $O(N)$

B) $O(N\log N)$

C) $O(N^2)$

D) $O(N^3)$

E) None of the above

```
public static int dp(int[] ints)
{
  int[] dp = new int[ints.length];
  Arrays.fill(dp, 1);
  int ret = 0;
  for(int i = 1;i<dp.length;i++)
  {
    for(int j = i-1;j>=0;j--)
    {
      if(ints[j] <= ints[i])
      {
        dp[i] = Math.max(dp[i],dp[j] + 1);
      }
    }
    ret = Math.max(ret, dp[i]);
  }
  return ret;
}

public static int rec(int[] ints, int ind)
{
  int r = 1;
  for(int i = ind+1;i<ints.length;i++)
  {
    if(ints[i] >= ints[ind])
    r = Math.max(r, 1 + rec(ints, i));
  }
  return r;
}
```

Question 25.

How many instance variables does a member of the `fish` class have?

A) 0

B) 1

C) 2

D) 3

E) 4

Question 26.

What is the output of the client code shown below?

```
pet a = new pet(5, "jeff");
pet b = new fish(5, "jeff");
out.println(a.equals(b));
```

A) true

B) false

C) jeff

D) There is no output due to a compile error

E) There is no output due to a runtime error

Question 27.

What is the output of the client code shown below?

```
pet a = new pet(5, "jeff");
pet b = new fish(5, "jeff");
out.println(b.equals(a));
```

A) true

B) false

C) jeff

D) There is no output due to a compile error

E) There is no output due to a runtime error

Question 28.

What is the output of the client code shown below?

```
pet a = new fish(5, "jeff");
pet b = new fish(5, "jeff");
out.println(a.equals(b));
```

A) true

B) false

C) jeff

D) There is no output due to a compile error

E) There is no output due to a runtime error

```
public class pet{

  private int weight;
  private String name;

  public pet(int w, String n)
  {
    this.weight = w;
    this.name = n;
  }

  public boolean equals(pet p)
  {
    if(weight == p.weight){
      return name.equals(p.name);
    }
    return false;
  }

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

}

private class fish extends pet{

  public fish(int w, String n)
  {
    super(w, n);
  }

  public boolean equals(pet p)
  {
    if(p instanceof fish)
      return p.equals(this);
    return false;
  }

}
```

Question 29.

What is the worst case time complexity of `add()` for Java's `PriorityQueue`? Assume comparison is an O(1) operation.

A) `O(1)`

B) `O(log(N))`

C) `O(N)`

D) `O(NlogN)`

E) $O(N^2)$

Question 30.

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

A) `0`

B) `6`

C) `108`

D) There is no output due to a compile error

E) There is no output due to a runtime error

```
int num = 27;
num >>= 34;
out.print(num);
```

Question 31.

What is the **best case** time complexity of bubble sort?

A) O(1)

B) O(N)

C) O(NlogN)

D) O(N^2)

E) None of the above

Question 32.

The method `run()` takes 2 seconds to complete on an array of 100 elements. Which of the following is the best estimate of how long it would take to complete on an array of 300 elements?

A) `2 seconds`

B) `6 seconds`

C) `18 seconds`

D) `36 seconds`

E) `54 seconds`

```
static int run(int[] ints){
   int ret = 0;
   for(int i: ints)
      for(int j: ints)
         for(int k: ints)
            ret += i+j+k;
   return ret;
}
```

Question 33.

Which of the following is logically equivalent to the digital electronics diagram to the right?

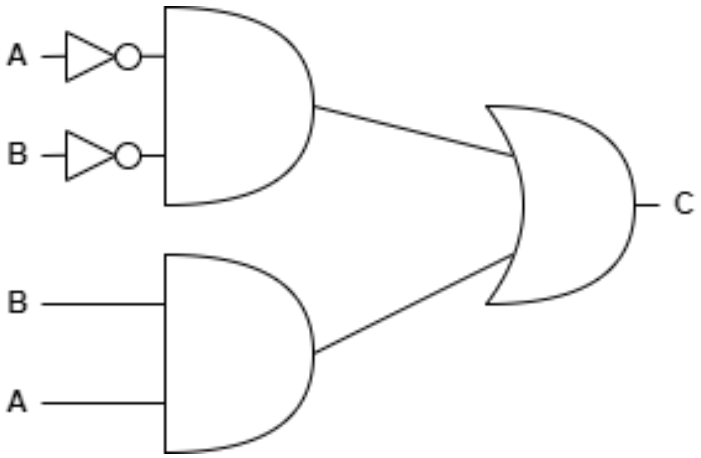
A) A ⊕ B

B) (A * B) + (!A * !B)

C) (A * B)

D) (A + B)

E) None of the above



Question 34.

What is the output of the client code below?

```
Inline obj = new Inline(){
   public int go(int x){return ~x;}
};
out.print(obj.go(27));
```

A) -26

B) -27

C) 27

D) -28

E) There is no output due to a compile error

```
public interface Inline
{
   public int go(int i);
}
```

Question 35.

The letters "DBFCAGE" are inserted into a binary search tree in order. Which of the following traversals will result in a visitation order of DBFACEG?

A) Pre-order traversal

B) Post-order traversal

C) In-order traversal

D) Level-order traversal

E) None of the above

For questions 36 – 38, use the structure code to the right and client code below:

```
Structure s = new Structure(10);
int[] a = new int[] {1,4,6,9,3,4,3};
int[] b = new int[] {5,8,7,2,7,6,8};
int y = 0;
int n = 0;
for(int i = 0;i<a.length;i++)
{
   boolean result = s.union(a[i], b[i]);
   if(result)
   {
      y++;
   }else {
      n++;
   }
}
/*LINE 1:*/
out.println(y+" "+n);
/*LINE 2:*/
out.println(s.find(3));
int mx = 0;
for(int i = 0;i<10;i++) {
   mx = Math.max(mx, s.sz(i));
}
/*LINE 3:*/
out.println(mx);
```

```
public class Structure{

  int[] p;

  public Structure(int sz)
  {
     p = new int[sz];
     Arrays.fill(p, -1);
  }

  public int find(int x)
  {
     if(p[x] < 0)
        return x;
     int ret = find(p[x]);
     p[x] = ret;
     return ret;
  }

  public int sz(int x)
  {
     return -p[find(x)];
  }

  public boolean union(int a, int b)
  {
     int pa = find(a);
     int pb = find(b);
     if(pa == pb) {
        return false;
     }
     p[pa] += p[pb];
     p[pb] = pa;
     return true;
  }

}
```

Question 36.

What is the output of the client code marked as **Line 1**?

A) 4 3

B) 5 2

C) 6 1

D) 7 0

E) None of the above

Question 37.

What is the output of the client code marked as **Line 2**?

A) 1

B) 3

C) 4

D) 7

E) 2

Question 38.

What is the output of the client code marked as **Line 3**?

A) 1

B) 2

C) 3

D) 5

E) 9

Question 39.

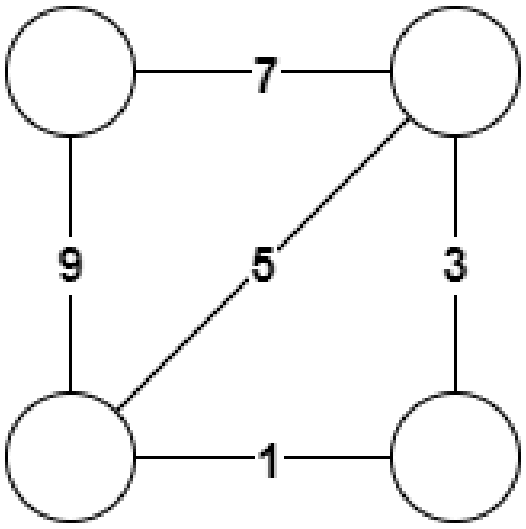
How many unique strings could replace ***CODE*** for the code below to output `true`? Write your answer in the blank provided on the answer document.

```
String A = "[ABC]{1,2}";
String B = "[BCD]{2,3}";
String C = \*CODE*\;
out.print(C.matches(A) && C.matches(B));
```

Question 40.

A minimum spanning tree is a subset of edges of minimum total weight that connect a graph. In other words, any two vertices in the graph can reach each other by traversing only edges selected to be part of the minimum spanning tree.

Below is a weighted, undirected graph. What is the total edge weight sum of a minimum spanning tree of the graph below? Write your answer in the blank provided on the answer document.



★ ANSWER KEY – CONFIDENTIAL ★

UIL COMPUTER SCIENCE – 2022 REGION

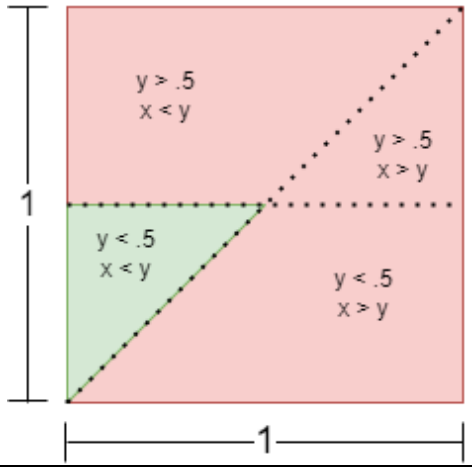
Questions (+6 points for each correct answer, -2 points for each incorrect answer)

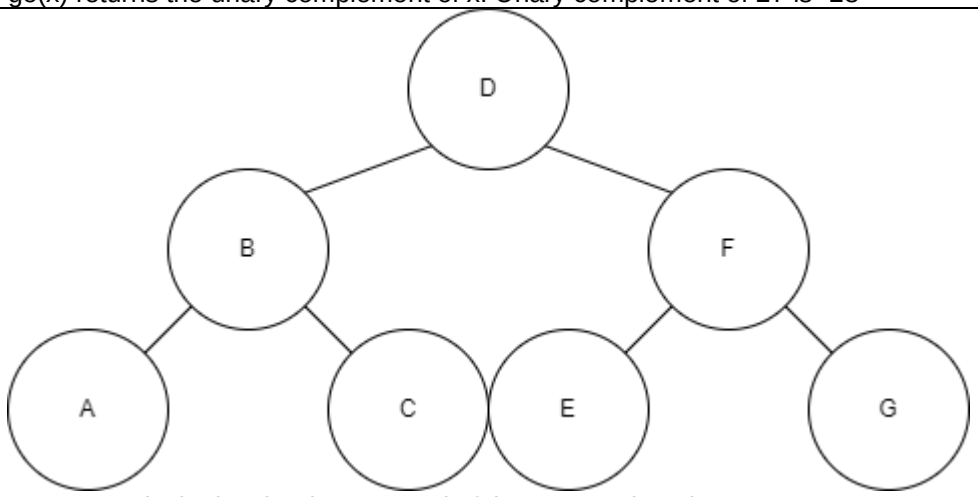
1) B	11) E	21) D	31) B
2) D	12) C	22) C	32) E
3) B	13) D	23) C	33) B
4) D	14) A	24) E	34) D
5) B	15) B	25) C	35) D
6) C	16) A	26) A	36) C
7) A	17) D	27) B	37) C
8) C	18) B	28) E	38) D
9) D	19) D	29) B	*39) 4
10) D	20) A	30) B	*40) 11

** 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.	B	101101 = 45, 10010 = 18, 45 * 18 = 810 = 1100101010
2.	D	(17 % 12 * 23) – (14 / 2) = (115) – (7) = 108
3.	B	"%%" formats to "%", and "%.2f" formats a floating point to two places after the decimal
4.	D	endsWith takes only one argument. The overload in the question does not exist.
5.	B	!(a || b) and (!a && !b) are logically equivalent by De Morgan's law, and two like values will always xor to false
6.	C	(27 * 27) = (3 * 3 * 3) (3 * 3 * 3) = (3 * 3) (3 * 3) (3 * 3) = 9^3
7.	A	The += shortcut operator truncates 4.5 to 4, and then 5 – 4 = 1
8.	C	Variable r starts as 1 and is then incremented by the "case" case and the default because no break is included above
9.	D	The arithmetic series 7, 11, 15, …, 83, 87 consists of (87 - 7)/4 + 1 = 21 terms. 2 asterisks are added per term, totaling to 42.
10.	D	The array resulting from these operations should be sorted and have a duplicate L and missing U
11.	E	FileNotFoundException, IOException, and Exception are all acceptable as each are more specific than the next.
12.	C	The sum of all powers of 2 from 1..n will be 2^n – 1.
13.	D	Unary complement has the highest precedence, followed by addition and then left shift
14.	A	The complement of Byte.MAX_VALUE is Byte.MIN_VALUE
15.	B	The 16'th element in the sequence will be 31, and the 8th element will be 15. 31-15 = 16
16.	A	A geometric intuition: x and y are selected uniformly from the square. The probability of the point landing in the green region is 1/8, .125 
17.	D	Primitives cannot be used in instanceof, compilation error
18.	B	The binary search code in this question will find the index of the weak floor (the largest value less than or equal to s). In this case that index is 4
19.	D	L = -1, R = 7, M = 3 L = 3, R = 7, M = 5 L = 3, R = 5, M = 4
20.	A	Both sets contain the same values, so equals() will evaluate to true. Because sets cannot contain duplicates, only 3 unique values will be present.
21.	D	Dp and rec both find the length of the longest weakly increasing subsequence. For #21, this is 1,1,1,1,7 -> 5
22.	C	3,4,5,9 or 1,1,5,9 -> 4
23.	C	There are O(n) elements to fill in the dp table, and each one takes O(n) time to fill, resulting in O(n^2) complexity
24.	E	The complexity of brute force recursive LIS is exponential, greater than all of the given answer choices
25.	C	Fish inherits 2 instance variables from pet
26.	A	Because a is a pet, the pet equals() method is called and returns true
27.	B	Because b is a fish, the fish equals() method is called and returns false

28.	E	Because both a and b are fish, they will call each others equals() method in an infinite loop, causing a runtime error
29.	B	PQ add is a log(n) operation
30.	B	X >> Y is actually evaluated as X >> ((Y % 32) + 32) % 32
31.	B	Bubble sort best case complexity occurs when a sorted array is given. In this case, bubble sort must look at each element once to verify that the array is sorted.
32.	E	Run() is an O(n^3) method. Hence, a 3x increase in array size will result in a 3^3 increase in runtime. 2*27 = 54
33.	B	this is a straightforward translation from digital electronics to logic
34.	D	go(x) returns the unary complement of x. Unary complement of 27 is -28
35.	D	 DBFACEG is the level order traversal of the tree produced.
36.	C	The structure in 36-38 is a Union-Find (also known as Disjoint Set) Union Find are used to maintain and join a collection of sets. Find(x) finds the leader of x's set Union(x,y) combines the sets x and y belong to, or returns false if they are already in the same set. This knowledge is not necessary to solve the problem though, it can be easily traced as well. For question 36, the only union operation to fail will be the last, as 3 and 8 already belong to the same set due to (4,8) (4,6) (6,7) (7,3). Hence y = 6 and n = 1
37.	C	The parent array of s before question 37 is `[-1, -2, 9, 4, -5, 1, 3, 6, 4, -2]`. Find(3) on this array will result in 4
38.	D	The maximum set size after all operations is 5
39.	4	C must be a string of length 2, consisting of only characters B and C. Valid strings are "BB","CC","BC","CB".
40.	11	The MST of the graph uses edge weights 1,3,7 (5 creates a cycle and would be useless, and 9 is more expensive).

Conference_____ Contestant Number_____

UIL COMPUTER SCIENCE WRITTEN TEST

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

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

FOR ADMINISTRATIVE USE ONLY

# Right:	×	6 pts	=	
# Wrong:	×	-2 pts	=	
# Skipped:	×	0 pts	=	0

	Score	Initials
Judge #1:		
Judge #2:		
Judge #3:		