

UIL COMPUTER SCIENCE WRITTEN TEST

2016 DISTRICT 1

MARCH 21-26, 2016

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(from, 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)
```

UIL COMPUTER SCIENCE WRITTEN TEST – 2016 DISTRICT 1

Note: Correct responses are based on **Java SE Development Kit 8 (JDK 8)** from Sun Microsystems, 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 8 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 equivalent to $100101_2 + 1111_2$?

- A) 101010_2 B) 65_8 C) 104_{10} D) 33_{16} E) $1G_{36}$

Question 2.

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

- A) 0 B) 3 C) 3.125 D) 4 E) 4.125

```
int x = 25;
int y = 8;
out.println((x + y) / y);
```

Question 3.

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

- A) 8573764954 B) 61LIU C) UIL16
D) 54 E) U
49 I
76 L
73 1
85 6

```
char[] uil = {'U', 'I', 'L', '1', '6'};
for (int i = uil.length - 1; i >= 0; i--)
    out.print(uil[i]);
```

Question 4.

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

- A) ransmog B) ansmogri C) ansmog
D) ransmo E) ansmogr

```
String calvin = "Transmogrifier";
out.println(calvin.substring(2, 8));
```

Question 5.

Which of the following Boolean expressions is equivalent to the truth table for output X, as shown to the right?

- A) $Q \ \&\& \ (!P \ || \ R)$
B) $Q \ || \ !(P \ \&\& \ !R)$
C) $(Q \ || \ P) \ \&\& \ !R$
D) $(Q \ \&\& \ !P) \ || \ R$
E) $!(!Q \ || \ !P) \ \&\& \ R$

P	Q	R	X
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

Question 6.

Which of the following outputs can never be produced by the code segment to the right?

- A) 16 B) 17 C) 20 D) 35 E) 36

```
int ran = 20;
int dumb = 16;
int num = (int)(Math.random() * ran + dumb);
out.println(num);
```

Question 7.

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

- A) -3.0 B) -3 C) 3 D) 3.3
E) No output due to an error.

```
int huey = -10;
int dewey = huey * 3;
int louie = dewey - 3;
out.println(louie % huey);
```

<p>Question 8.</p> <p>What is the output of the code segment to the right if the value of test is initialized as follows?</p> <pre>int test = 5;</pre> <p>A) 5 B) five.5 C) five.6 D) six.eight.9 E) five.six.eight.9</p>	<pre>switch(test) { case 5: out.print("five."); test++; break; case 6: out.print("six."); test += 3; case 8: out.print("eight."); } out.println(test);</pre>
<p>Question 9.</p> <p>What is the output of the code segment to the right?</p> <p>A) @@@ B) @@@@@@ C) @@@@@@ D) @@@@@@@@ E) The code segment prints an infinite string of @ characters.</p>	<pre>int at = 3; do { out.print("@"); at *= 3; } while (at < 2500);</pre>
<p>Question 10.</p> <p>What is the return value of the following invocation of the process () method from a client class?</p> <pre>int[] auto = {13, 9, 3, 11, 5, 12}; out.println(process(auto));</pre> <p>A) 0 B) 17 C) 33 D) 40 E) 53</p>	<pre>public static int process(int[] a) { int b = 0; int c = 0; for (int d : a) { if (d < b) c += d; b = d; } return c; }</pre>
<p>Question 11.</p> <p>What is the output of the code segment to the right if the user enters the following line of console input?</p> <pre>28 11 30 15 33 45 END</pre> <p>A) 13 11 8 15 3 0 B) 15 4 4 4 4 4 C) 15 6 6 6 6 6 D) 13 11 0 0 3 0 E) 13 11 8 7 5 0</p>	<pre>Scanner user = new Scanner(System.in); int alfa = 15; while (user.hasNextInt()) { int bravo = user.nextInt(); alfa = bravo % alfa; out.print(alfa + " "); }</pre>
<p>Question 12.</p> <p>What is the output of the code segment to the right?</p> <p>A) 994 B) 1000 C) 1994 D) 2000 E) 3992</p>	<pre>int half = 0; for (int i = 1000; i > 0; i /= 2) half += i; out.println(half);</pre>
<p>Question 13.</p> <p>What is the output of the code segment to the right?</p> <p>A) 0 B) 5 C) 23 D) 55 E) 1472</p>	<pre>out.println(23 6 << 3 ^ 7 & 13);</pre>
<p>Question 14.</p> <p>Which of the following abstract data types would be the most optimal choice for implementing the database for a mobile app that allows a customer to look up the price of a product by scanning its barcode (UPC)?</p> <p>A) LinkedList B) Stack C) HashMap D) TreeSet E) PriorityQueue</p>	
<p>Question 15.</p> <p>What is the output of the code segment to the right?</p> <p>A) [0, 1, 3, 4, 24, 30, 36, 42, 48, 54] B) [0, 6, 12, 6, 24, 30, 12, 42, 48, 18] C) [0, 1, 1, 2, 3, 30, 36, 42, 48, 54] D) [0, 1, 12, 2, 3, 30, 36, 42, 48, 54] E) [0, 6, 12, 18, 24, 30, 36, 42, 48, 54]</p>	<pre>List<Byte> bytes = new ArrayList<Byte>(); for (int i = 0; i < 10; i++) { bytes.add((byte)(i * 6)); if (i % 3 == 0) bytes.set((byte)(i / 2), (byte)(i / 3)); } out.println(bytes);</pre>

<p>Question 16.</p> <p>What is printed by the following invocation of the <code>findX()</code> method from a client class?</p> <pre>out.println(findX(9));</pre> <p>A) 04 B) 34 C) 09 D) 39 E) 123-1</p>	<pre>public static int findX(int x) { int i = -1; int code = 0; int[] a = {7, 1, 3, 4, 9, 8, 2, 5, 0, 9}; try { while (a[++i] != x) {} } catch (RuntimeException e) { code = code * 10 + 1; } catch (Exception e) { code = code * 10 + 2; } finally { code = code * 10 + 3; } out.print(code); return i; }</pre>
<p>Question 17.</p> <p>What is printed by the following invocation of the <code>findX()</code> method from a client class?</p> <pre>out.println(findX(6));</pre> <p>A) 1310 B) 123-1 C) 2310 D) 210 E) 010</p>	
<p>Question 18.</p> <p>What is the output of the code segment to the right?</p> <p>A) This B) That C) the Other D) null E) No output due to an error.</p>	<pre>String[] label = {"This", "That", "the Other" }; if (label[0].compareTo(label[1]) < 0) if (label[0].compareTo(label[2]) > 0) out.println(label[0]); else out.println(label[2]); else if (label[2].compareTo(label[1]) < 0) out.println(label[1]); else out.println("null");</pre>
<p>Question 19.</p> <p>What is the output of the code segment to the right?</p> <p>A) 10 B) 14 C) 19 D) 24 E) No output due to an error.</p>	<pre>out.println(Integer.toString(19, 15));</pre>
<p>Question 20.</p> <p>Which of the following strings does NOT match the regular expression shown to the right?</p> <p>A) aaabbbaba B) aabbaabba C) abba D) abbbabbbba E) abaabbaaba</p>	<p style="text-align: center;">a+bb(a*b)*a</p>
<p>Question 21.</p> <p>What is printed by the following invocation of the <code>mangle()</code> method from a client class?</p> <pre>out.println(mangle("DistrictUIL"));</pre> <p>A) iDistrctUIL B) isiDrtUtcLI C) DitrscitILUi D) iULItcsrtiD E) ILctUtrDisi</p>	<pre>public static String mangle(String s) { if (s.length() == 0) return ""; int i = s.length() / 2; char c = s.charAt(i); String s1 = s.substring(0, i); String s2 = s.substring(i+1); return c + mangle(s1) + mangle(s2); }</pre>
<p>Question 22.</p> <p>What is the output of the code segment to the right?</p> <p>A) 42.43. B) 7. 63.64.65. 9.10.11. 84.85.86.87. 11.12.13.14. C) 43. D) 6.7. 63.64.65. 9.10.11. 83.84.85.86. 12.13.14.15. E) 7.9.10.11.11.12.13.14.</p>	<pre>for (int six = 4; six < 10; six += 2) { for (int two = six / 2; two < six; two++) out.print(six + two + "."); out.println(); }</pre>

Question 23.

Which of the following algorithms is implemented by the `alg()` method to the right?

- A) Selection Sort B) Binary Search
C) Merge Sort D) Insertion Sort
E) Quicksort

Question 24.

What is the expected runtime performance for the `alg()` method in the best case? Choose the most restrictive answer.

- A) $O(N)$ B) $O(N^2)$ C) $O(\log_2 N)$
D) $O(N * \log_2 N)$ E) Indeterminate

Question 25.

What is the output of line <#1> in the **Client Code** to the right?

- A) -1 B) 0 C) 7 D) 8 E) 28

Question 26.

What is the output of line <#2> in the **Client Code** to the right?

- A) District
B) [D, c, i, i, r, s, t, t]
C) ttsriicD
D) [D, i, s, t, r, i, c, t]
E) [t, t, s, r, i, i, c, D]

Question 27.

Given the code segment to the right, how many of the elements of the `chunks` array are empty strings?

- A) 0 B) 1 C) 2 D) 3 E) 5

Question 28.

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

- A) bceeg
B) bdddff
C) abcdefg
D) aabbcddefffg
E) No output due to an error.

Question 29.

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

- A) 0 B) 5 C) 10 D) 84 E) 3125

Question 30.

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

- A) [1, 5, 8, 10]
B) [5, 6, 9]
C) [1, 2, 3, 4]
D) [2, 6, 7]
E) No output due to an error.

```
static int alg(char[] a) {
    int n = -1;
    for (int i = 0; i < a.length; i++) {
        int m = i;
        for (int j = a.length-1; j > i; j--) {
            if (a[m] < a[j])
                m = j;
        }
        help(a, i, m);
        n++;
    }
    return n;
}
```

```
static void help(char[] a, int x, int y){
    char z = a[x];
    a[x] = a[y];
    a[y] = z;
}
```

Client Code

```
String dist = "District";
char[] data = dist.toCharArray();
out.println(alg(data));      //<#1>
String s = Arrays.toString(data);
out.println(s);              //<#2>
```

```
String block = "to be or not to be";
String axe = ".\\s.";
String[] chunks = block.split(axe);
```

```
Queue<Object> que = new LinkedList<>();
String pattern = "aabbcddefffg";
for (char c : pattern.toCharArray())
    que.add(c);
while (!que.isEmpty()) {
    if (!que.remove().equals(que.peek()))
        out.print(que.remove());
}
```

```
out.println(2 + 3 ^ 4 + 1);
```

```
int[][] grid = { {1, 2, 3, 4}, {5, 6, 7},
                 {8, 9}, {10} };
for (int i = 0; i < grid.length; i++)
    for (int j = 0; j < grid[i].length; j++)
        grid[i][j] = grid[j][i];
out.println(Arrays.toString(grid[1]));
```

Question 31.

Given the class and interface definitions to the right, what is the output of the following code segment?

```
Customer moe = new Cheapo();
moe.pay(5.00);
moe.pay(2.00);
moe.pay(10.50);
out.println(moe);
```

- A) \$17.5 + \$0.0
- B) \$17.5 + \$3.5
- C) \$17.5 + \$8.75
- D) \$10.5 + \$0.0
- E) No output due to an error.

```
public interface Customer {
    public double rate = 1.20;
    public double pay(double amount);
}
```

```
public class Cheapo implements Customer {
    public double amounts;
    public double tips;
    public double rate = 0.00;
```

```
    public Cheapo() {}
```

```
    public double pay(double amount) {
        amounts += amount;
        double tip = amount * rate;
        tips += tip;
        return amount + tip;
    }
```

```
    public String toString() {
        return "$" + amounts + " + $" + tips;
    }
}
```

```
public class MoneyBags extends Cheapo {
    public double rate = 1.50;
```

```
    public double pay(double amount) {
        return super.pay(amount);
    }
```

```
    private String getRate() {
        return (int)((rate - 1) * 100) + "%";
    }
```

```
    public String toString() {
        return super.toString() + " (" +
            getRate() + ")";
    }
}
```

Question 32.

Given the class and interface definitions to the right, what is the output of the following code segment?

```
Customer larry = new MoneyBags();
larry.pay(5.00);
larry.pay(2.00);
larry.pay(10.50);
out.println(larry);
```

- A) \$17.5 + \$0.0 (0%)
- B) \$17.5 + \$0.0 (50%)
- C) \$17.5 + \$8.75 (50%)
- D) \$17.5 + \$3.5 (20%)
- E) No output due to an error.

Question 33.

Given the class and interface definitions to the right, what is the output of the following code segment?

```
Customer curly = new Customer();
curly.pay(5.00);
curly.pay(2.00);
curly.pay(10.50);
out.println(curly);
```

- A) \$17.5 + \$0.0
- B) \$17.5 + \$3.5
- C) \$17.5 + \$8.75
- D) \$10.5 + \$0.0
- E) No output due to an error.

```
Map<String, String> map = new HashMap<>();
map.put("one", "two");
map.put("two", "two");
map.put("four", "five");
map.put("five", "one");
map.put("one", "four");
out.println(map.get("one") + " " +
    map.get("two") + " " +
    map.get("three"));
```

Question 34.

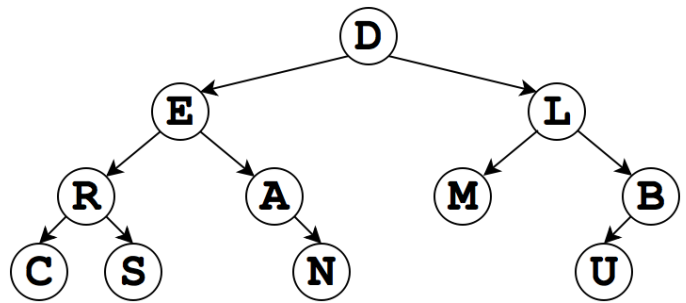
What is the output of the code segment to the right?

- A) five [one, two]
- B) [two, four] two
- C) five two null
- D) four two null
- E) five two

Question 35.

Which of the following is a post-order traversal of the binary tree to the right?

- A) CRSEANDMLUB
 B) DELRAMBCSNU
 C) CSRNAEMUBLD
 D) DERCSANLMBU
 E) CRSEANMLUBD

**Question 36.**

Given the truth table to the right with inputs P, Q, and R, which of the following is a valid statement about output X?

- A) X will always be 0 whenever the values of P and R are different from each other.
 B) X will always be 1 whenever P is 1.
 C) X will always be 1 whenever exactly 2 inputs are both 1.
 D) X will always be 0 whenever Q is 0.
 E) More than one of these statements is valid.

P	Q	R	X
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Question 37.

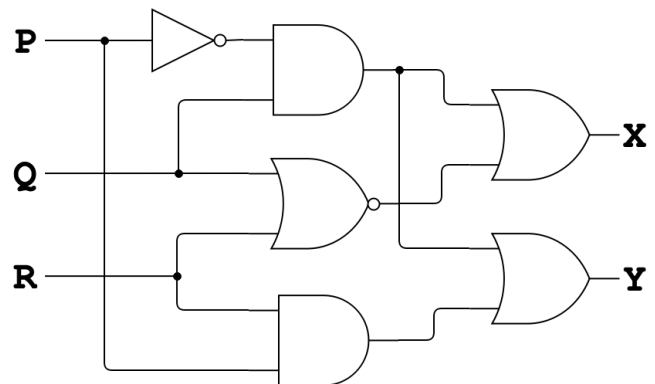
What is the 8-bit, 2's complement representation of -100_{10} ?

- A) 10011011_2 B) -00000100_2 C) 11100100_2 D) 10011100_2 E) 10000100_2

Question 38.

Which of the following set of inputs for the logic diagram to the right will result in a false outputs for both X and Y?

- A) P = false; Q = false; R = false
 B) P = false; Q = false; R = true
 C) P = true; Q = false; R = false
 D) P = true; Q = false; R = true
 E) P = true; Q = true; R = true

**Question 39.**

Write a simplified, Boolean expression that is equivalent to the expression to the right. Your answer should include as few logical operators as possible.

$$(A * B) + (C * B)$$

Write your answer on the answer sheet.

Question 40.

What is the postfix notation (reverse Polish notation) for the arithmetic expression to the right?

$$7 - (3 * 4)$$

Write your answer on the answer sheet.

★ DOUBLE-CHECK YOUR ANSWERS ★