

QUESTION 1	
<p>What does 1001_2 plus 1110_2 equal?</p> <p>A. 10111_2 B. 1001_2 C. 11111_2 D. 111_2 E. 32_{10}</p>	
<p>QUESTION 2</p> <p>What is output by the code to the right?</p> <p>A. 9 B. 10 C. 12</p> <p>D. 3 E. xyx</p>	<pre>int x = 3; int y = 2; System.out.println(x + y * x);</pre>
<p>QUESTION 3</p> <p>What is output by the code to the right?</p> <p>A. 10 B. 5 C. 0</p> <p>D. 6 E. 12</p>	<pre>int total = 0; for(int i = 0; i <= 5; i++){ total += 2; } System.out.println(total);</pre>
<p>QUESTION 4</p> <p>What is output by the code to the right?</p> <p>A. SOUTH B. SOUTH88 C. SOUTH**</p> <p>D. South88 E. SOUTH+**</p>	<pre>String s = "South88"; System.out.println(s.toUpperCase());</pre>
<p>QUESTION 5</p> <p>What is output by the code to the right?</p> <p>A. 7 B. 3</p> <p>C. 2 D. 5</p> <p>E. 1</p>	<pre>int[] data = {3, 2, 4, 3, 1, 0}; data[1] = data[1] + data[3]; System.out.println(data[1]);</pre>
<p>QUESTION 6</p> <p>What is output by the code to the right?</p> <p>A. 0 B. 20 C. 0.3</p> <p>D. 120 E. 6</p>	<pre>int r = 6; int v = 20; System.out.println(r % v);</pre>
QUESTION 7	
<p>Which answer is logically equivalent to the following boolean expression, where p and q are boolean variables?.</p> <p style="padding-left: 40px;">$p \ \&\& \ !q$</p> <p>A. $p \ \ !q$ B. $!p \ \&\& \ q$ C. $!(!p \ \ q)$ D. $!p \ \ q$ E. $!(!p \ \&\& \ q)$</p>	

<p>QUESTION 8</p> <p>What is output by the code to the right?</p> <p>A. 21 B. 2 C. 1</p> <p>D. 12 E. There is no output.</p>	<pre>double a = 2.5; double b = 15.7; if(a < b) System.out.print(1); if(b > 10) System.out.print(2);</pre>
<p>QUESTION 9</p> <p>What replaces <code><*1></code> in the code to the right to indicate that the method <code>takeTrip</code> does not return a value?</p> <p>A. return</p> <p>B. null</p> <p>C. static</p> <p>D. private</p> <p>E. void</p>	<pre>public class Car{ private int miles; public Car(int m){ miles = m; } public <*1> takeTrip(int len){ miles += len; } public int getDistance(){ return miles; } }</pre>
<p>Assume <code><*1></code> is filled in correctly.</p>	
<p>QUESTION 10</p> <p>Which of the following creates a <code>Car</code> object whose <code>miles</code> instance variable is initialized to zero?</p> <p>A. <code>Car c = new Car("0");</code></p> <p>B. <code>Car c = new Car('0');</code></p> <p>C. <code>Car c = new Car(0);</code></p> <p>D. <code>Car c = new Car(miles.0);</code></p> <p>E. <code>Car c = new Car("zero");</code></p>	
<p>QUESTION 11</p> <p>What is output by the code to the right?</p> <p>A. 13 B. 11 C. 2</p> <p>D. 9 E. 0</p>	<pre>int z = 2; int k = 11; System.out.print(k & z);</pre>
<p>QUESTION 12</p> <p>How many lines of output does the code to the right produce?</p> <p>A. 0 B. 1 C. 2</p> <p>D. 3 E. 4</p>	<pre>System.out.print("first string"); System.out.print("second string"); System.out.println("third string");</pre>
<p>QUESTION 13</p> <p>What is output by the code to the right?</p> <p>A. 7.0 B. 14 C. 7</p> <p>D. 14.0 E. 2</p>	<pre>System.out.println(Math.min(14, 7));</pre>

<p>QUESTION 14</p> <p>What is output by the code to the right?</p> <p>A. 0019 B. 19.0 C. 000019</p> <p>D. 19.00 E. 19</p>	<pre>System.out.printf("%04d", 19);</pre>
<p>QUESTION 15</p> <p>What is returned by the method call <code>simple(3)</code>?</p> <p>A. 6 B. 3 C. 10</p> <p>D. 8 E. 0</p>	<pre>public static int simple(int x){ x++; return x + x; }</pre>
<p>QUESTION 16</p> <p>What is output by the code to the right?</p> <p>A. 2 B. 4 C. 5</p> <p>D. There is no output due to a syntax error.</p> <p>E. There is no output due to an <code>ArrayIndexOutOfBoundsException</code>.</p>	<pre>String names = "Bob Don J Tim"; String[] chopped = names.split("\\s+"); System.out.print(chopped.length);</pre>
<p>QUESTION 17</p> <p>What is returned by the method call <code>rec(4)</code>?</p> <p>A. 4 B. 1 C. 24</p> <p>D. 10 E. -1</p>	<pre>public static int rec(int x){ if(x <= 1) return 1; else return x + rec(x - 1); }</pre>
<p>QUESTION 18</p> <p>What is output by the code to the right when method <code>two</code> is called?</p> <p>A. 3 B. 4 C. 1</p> <p>D. There is no output due to a syntax error.</p> <p>E. There is no output due to a runtime error.</p>	<pre>public static int one(int x){ return x + x; } public static int one(int x, int y){ return x + y; } public static void two(){ System.out.print(one(2, 1)); }</pre>
<p>QUESTION 19</p> <p>What is output by the code to the right?</p> <p>A. true grace B. true false</p> <p>C. true true D. false false</p> <p>E. false true</p>	<pre>Object obj = new Object(); String str = "grace"; System.out.print(obj instanceof String); System.out.print(" "); System.out.print(str instanceof Object);</pre>

<p>QUESTION 20</p> <p>What is output by the code to the right?</p> <p>A. false B. true C. door</p> <p>D. There is no output due to a syntax error.</p> <p>E. There is no output due to a runtime error.</p>	<pre>String item = "door"; System.out.print(item.matches("d..r"));</pre>
<p>QUESTION 21</p> <p>What is output by the code to the right?</p> <p>A. [3, 7] B. [7, 3] C. [3]</p> <p>D. [7, 0, 3] E. [0, 3, 7]</p>	<pre>ArrayList<Integer> nums = new ArrayList<Integer>(); nums.add(7); nums.add(0, 3); System.out.print(nums);</pre>
<p>QUESTION 22</p> <p>Which of the following could replace <*1> in the code to the right as a syntactically legal identifier?</p> <p>A. value B. int</p> <p>C. x+y D. num12</p> <p>E. More than one of these.</p>	<pre>int <*1> = 42;</pre>
<p>QUESTION 23</p> <p>The code to the right contains a syntax error. Which of the following best describes the reason for the syntax error?</p> <p>A. Duplicates may not be added to a Set.</p> <p>B. "B" is a char, not a String.</p> <p>C. Instances of interfaces cannot be created.</p> <p>D. Sets cannot be iterated over using the enhanced for loop.</p> <p>E. Sets cannot contain Strings.</p>	<pre>Set<String> smallSet = new Set<String>(); smallSet.add("A"); smallSet.add("B"); smallSet.add("A"); for(String str : smallSet) System.out.print(str);</pre>
<p>QUESTION 24</p> <p>What is output by the code to the right?</p> <p>A. X B. Y C. Z</p> <p>D. ZY E. YX</p>	<pre>Queue<String> q = new LinkedList<String>(); q.add("Z"); q.add("X"); q.add("Y"); System.out.print(q.remove());</pre>
<p>QUESTION 25</p> <p>What is output by the code to the right?</p> <p>A. 8 B. 0 C. 6</p> <p>D. 7 E. 5</p>	<pre>int[] ary = {5, 7, 3}; int[] otherAry = ary; otherAry[1]++; otherAry = new int[5]; System.out.print(ary[1]);</pre>

<p>QUESTION 26</p> <p>How many *'s are output by the code to the right?</p> <p>A. 27 B. 3 C. 10</p> <p>D. 30 E. 13</p>	<pre>for(int i = 0; i < 10; i++) for(int j = 0; j < 3; j++) System.out.print("*");</pre>
<p>QUESTION 27</p> <p>What replaces <*1> in the code to the right so that if the element at index j is less than the element at index temp according to their natural ordering, the statement temp = j; is executed?</p> <p>A. temp.compareTo(j) <= 0</p> <p>B. data[j] < data[temp]</p> <p>C. data[j].compareTo(data[temp]) == 0</p> <p>D. j.compareTo(data[temp]) > 0</p> <p>E. data[j].compareTo(data[temp]) < 0</p>	
<p>Assume <*1> is filled in correctly.</p>	
<p>QUESTION 28</p> <p>What replaces <*2> in the code to the right so that the elements originally at indices i and j in array data are swapped with each other?</p> <p>A. int t = i; i = j; j = t;</p> <p>B. Comparable t = data[i]; data[i] = data[j]; data[j] = t;</p> <p>C. data[i] = data[i] ^ data[j]; data[j] = data[i] ^ data[j]; data[i] = data[j] ^ data[i];</p> <p>D. data[i] = data[j]; data[j] = data[i];</p> <p>E. More than one of these.</p>	<pre>public static void sort(Comparable[] data) { int temp; int len = data.length; for(int i = 0; i < len - 1; i++){ temp = i; for(int j = i + 1; j < len; j++){ if(<*1>) temp = j; } swap(data, i, temp); } } public static void swap(Comparable[] data, int i, int j){ <*2> }</pre>
<p>Assume <*1> and <*2> are filled in correctly.</p>	
<p>QUESTION 29</p> <p>What sorting algorithm is implemented by methods sort and swap?</p> <p>A. Insertion sort B. Quick Sort</p> <p>C. Selection Sort D. Shell Sort</p> <p>E. Merge Sort</p>	

<p>QUESTION 30</p> <p>What replaces <code><*1></code> in the code to the right to indicate that the <code>TreeMap</code> named <code>encode</code> has <code>Strings</code> for keys and <code>Integers</code> for values?</p> <p>A. <code><Integer, String></code> B. <code><String, int></code> C. <code><int, String></code> D. <code><String><int></code> E. <code><String, Integer></code></p>	<pre>TreeMap<*1> encode = new TreeMap<*1>(); encode.put("M", 212); encode.put("A", 193); encode.put("T", 227);</pre>
<p>Assume <code><*1></code> is filled in correctly.</p>	<pre>Iterator< Map.Entry<*1> > it; it = encode.entrySet().iterator(); System.out.print(it.next().getValue());</pre>
<p>QUESTION 31</p> <p>What is output by the code to the right?</p> <p>A. 193 B. M C. A D. T E. 227</p>	
<p>QUESTION 32</p> <p>What is output by the code to the right when method <code>first</code> is called?</p> <p>A. 1 B. 0 C. 2 D. 5 E. There is no output due to a runtime error.</p>	<pre>/* pre: data != null, elements of data are sorted in ascending order. */ public static int find(int tgt, int[] data){ int en = data.length - 1; return help(0, en, tgt, data); } private static int help(int st, int en, int tgt, int[] data){ int result = -1; int md, val; if(st <= en){ md = (st + en) / 2; val = data[md]; if(val == tgt) result = md; else if(tgt < val) result = help(st, md - 1, tgt, data); else result = help(md + 1, en, tgt, data); } return result; }</pre>
<p>QUESTION 33</p> <p>What searching algorithm is implemented by methods <code>find</code> and <code>help</code>?</p> <p>A. linear search B. interpolation search C. random search D. comb search E. binary search</p>	
<p>QUESTION 34</p> <p>Given an array that contains <code>N</code> elements what is the expected running time of method <code>find</code>? Choose the most restrictive correct answer.</p> <p>A. <code>O(N)</code> B. <code>O(1)</code> C. <code>O(logN)</code> D. <code>O(NlogN)</code> E. <code>O(sqrt(N))</code></p>	<pre>public static void first(){ int[] data = {0, 5, 19, 100}; System.out.print(find(5, data)); }</pre>

QUESTION 35

What replaces `<*1>` in the code to the right so that method `isEmpty` returns true if the `ArrayList myCon` contains 0 elements?

- A. `myCon.size() == 0 ? false : true`
- B. `return size() > 0;`
- C. `return super.size() == 0`
- D. `return myCon.size() == 0`
- E. `super.myCon.isEmpty();`

Assume `<*1>` is filled in correctly.

QUESTION 36

What is output by the code to the right when method `second` is called?

- A. CBA B. ABC C. CB
- D. C E. CCC

QUESTION 37

What type of data structure does the `Structure` class implement?

- A. List B. Stack C. Queue
- D. Heap E. Binary Search Tree

```
public class Structure<E>{
    private ArrayList<E> myCon;

    public Structure(){
        myCon = new ArrayList<E>();
    }

    public void add(E obj){
        myCon.add(obj);
    }

    public E peek(){
        return myCon.get( myCon.size() - 1 );
    }

    public boolean isEmpty(){
        <*1>;
    }

    public E remove(){
        return myCon.remove(myCon.size() - 1);
    }
}

////////// client code //////////
public static void second(){
    Structure<String> s
        = new Structure<String>();
    s.add( "A" );
    s.add( "B" );
    s.add( "C" );
    while( !s.isEmpty() )
        System.out.print( s.remove() );
}
```

QUESTION 38

Assume the method `sample(int[] data)` is $O(N^2)$ where $N = data.length$. When the method `sample` is passed an array with `length = 100,000` it takes 2 seconds for method `sample` to complete. If method `sample` is then passed an array with `length = 200,000` what is the expected time it will take method `sample` to complete?

- A. 2 seconds B. 3 seconds C. 4 seconds D. 6 seconds E. 8 seconds

QUESTION 39

The following values are inserted in the order shown into a binary search tree using the traditional insertion algorithm. What is the result of a post order traversal of the resulting tree?

2, 6, 1, 8, 0

- A. 2 1 0 6 8 B. 0 1 2 6 8 C. 0 1 8 6 2 D. 2 1 6 0 8 E. 0 8 1 6 2

QUESTION 40

Which keyword is used in a method declaration to indicate the method may generate an exception, but will not try to handle it locally?

- A. try B. throws C. catch D. throw E. finally

Standard Classes and Interfaces — Supplemental Reference

class java.lang.Object

- o boolean equals(Object other)
- o String toString()
- o int hashCode()

interface java.lang.Comparable<T>

- o int compareTo(T other)
Return value < 0 if this is less than other.
Return value = 0 if this is equal to other.
Return value > 0 if this is greater than other.

class java.lang.Integer implements Comparable<Integer>

- o Integer(int value)
- o int intValue()
- o boolean equals(Object obj)
- o String toString()
- o int compareTo(Integer anotherInteger)
- o static int parseInt(String s)

class java.lang.Double implements Comparable<Double>

- o Double(double value)
- o double doubleValue()
- o boolean equals(Object obj)
- o String toString()
- o int compareTo(Double anotherDouble)
- o static double parseDouble(String s)

class java.lang.String implements Comparable<String>

- o int compareTo(String anotherString)
- o boolean equals(Object obj)
- o int length()
- o String substring(int begin, int end)
Returns the substring starting at index begin and ending at index (to-1).
- o String substring(int begin)
Returns substring(from, length()).
- o int indexOf(String str)
Returns the index within this string of the first occurrence of the specified substring. Returns -1 if str is not found.
- o int indexOf(String str, int fromIndex)
Returns the index within this string of the first occurrence of the specified substring, starting the search at the specified index. Returns -1 if str is not found.
- o charAt(int index)
- o int indexOf(int ch)
- o int indexOf(int ch, int fromIndex)
- o String toLowerCase()
- o String toUpperCase()
- o String[] split(String regex)
- o boolean matches(String regex)

class java.lang.Character

- o static boolean isDigit(char ch)
- o static boolean isLetter(char ch)
- o static boolean isLetterOrDigit(char ch)
- o static boolean isLowerCase(char ch)
- o static boolean isUpperCase(char ch)
- o static char toUpperCase(char ch)
- o static char toLowerCase(char ch)

class java.lang.Math

- o static int abs(int a)
- o static double abs(double a)
- o static double pow(double base, double exponent)
- o static double sqrt(double a)
- o static double ceil(double a)
- o static double floor(double a)
- o static double min(double a, double b)
- o static double max(double a, double b)
- o static int min(int a, int b)
- o static int max(int a, int b)
- o static long round(double a)
- o static double random()
Returns a double value with a positive sign, greater than or equal to 0.0 and less than 1.0.

interface java.util.List<E>

- o boolean add(E e)
- o int size()
- o Iterator<E> iterator()
- o ListIterator<E> listIterator()

class java.util.ArrayList<E> implements List<E>

- Methods in addition to the List methods:
- o E get(int index)
 - o E set(int index, E e)
Replaces the element at index with x.
 - o void add(int index, E e)
Inserts x at position index, sliding elements at position index and higher to the right (adds 1 to their indices) and adjusts size.
 - o E remove(int index)
Removes element from position index, sliding elements at position (index + 1) and higher to the left (subtracts 1 from their indices) and adjusts size.

class java.util.LinkedList<E> implements List<E>, Queue<E>

- Methods in addition to the List methods:
- o void addFirst(E e)
 - o void addLast(E e)
 - o E getFirst()
 - o E getLast()
 - o E removeFirst()
 - o E removeLast()

class java.util.Stack<E>

- o boolean isEmpty()
- o E peek()
- o E pop()
- o E push(E item)

interface java.util.Queue<E>

- o boolean add(E e)
- o boolean isEmpty()
- o E peek()
- o E remove()

class java.util.PriorityQueue<E>

- o boolean add(E e)
- o boolean isEmpty()
- o E peek()
- o E remove()

interface java.util.Set<E>

- o boolean add(E e)
- o boolean contains(Object obj)
- o boolean remove(Object obj)
- o int size()
- o Iterator<E> iterator()
- o boolean addAll(Collection<? extends E> c)
- o boolean removeAll(Collection<?> c)
- o boolean retainAll(Collection<?> c)

class java.util.HashSet<E> implements Set<E>

class java.util.TreeSet<E> implements Set<E>

interface java.util.Map<K,V>

- o Object put(K key, V value)
- o V get(Object key)
- o boolean containsKey(Object key)
- o int size()
- o Set<K> keySet()
- o Set<Map.Entry<K, V>> entrySet()

class java.util.HashMap<K,V> implements Map<K,V>

class java.util.TreeMap<K,V> implements Map<K,V>

interface java.util.Map.Entry<K,V>

- o K getKey()
- o V getValue()
- o V setValue(V value)

interface java.util.Iterator<E>

- o boolean hasNext()
- o E next()
- o void remove()

interface java.util.ListIterator<E> extends

java.util.Iterator<E>

Methods in addition to the Iterator methods:

- o void add(E e)
- o void set(E e)

class java.lang.Exception

- o Exception()
- o Exception(String message)

class java.util.Scanner

- o Scanner(InputStream source)
- o boolean hasNext()
- o boolean hasNextInt()
- o boolean hasNextDouble()
- o String next()
- o int nextInt()
- o double nextDouble()
- o String nextLine()
- o Scanner useDelimiter(String pattern)

Computer Science Answer Key

UIL Invitational A 2008

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

Notes:

22. Choices A and D are both syntactically legal identifiers.

31. The `TreeMap` stores keys in ascending order, thus the first entry in the map will be ["A", 193] and "A" is the key for that entry.

The clause "Choose the most restrictive correct answer." is necessary because per the formal definition of Big O, an algorithm that is $O(N^2)$ is also $O(N^3)$, $O(N^4)$, and so forth.