

University Interscholastic League

Computer Science Competition

Number 128 (District 2 - 2011)

General Directions (Please read carefully!):

- 1) DO NOT OPEN EXAM UNTIL TOLD TO DO SO.
- 2) **NO CALCULATOR OF ANY KIND MAY BE USED.**
- 3) There are 40 questions on this contest exam. You have 45 minutes to complete this contest. If you are in the process of actually writing an answer when the signal to stop is given, you may finish writing that answer.
- 4) Papers 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 paper until told to do otherwise. Use this time to check your answers.
- 5) All answers must be written on the answer sheet/Scantron card provided. Indicate your answers in the appropriate blanks provided on the answer sheet or on the Scantron card. Clean erasures are necessary for accurate Scantron grading.
- 6) You may place as many notations as you desire anywhere on the test paper, but not on the answer sheet or Scantron card which are reserved for answers only.
- 7) You may use additional scratch paper provided by the contest director.
- 8) All questions have ONE and only ONE correct (BEST) answer. There is a penalty for all incorrect answers. **All provided code segments are intended to be syntactically correct, unless otherwise stated. Ignore any typographical errors and assume any undefined variables are defined as used.**
- 9) A reference to commonly used Java classes is provided at the end of the test, and you may use this reference sheet during the contest. You may detach the reference sheets from the test booklet, but DO NOT DO SO UNTIL THE CONTEST BEGINS.
- 10) Assume that any necessary import statements for standard Java packages and classes (e.g. `.util`, `ArrayList`, etc.) are included in any programs or code segments that refer to methods from these classes and packages.

Scoring:

- 1) All questions will receive **6 points** if answered correctly; no points will be given or subtracted if unanswered; **2 points** will be deducted for an incorrect answer.

QUESTION 1

What is the sum of $A56_{16}$ and $2D6_{16}$?

- A. $D32_{16}$ B. $C3B_{16}$ C. $D2C_{16}$ D. $B2B_{16}$ E. $1A12_{16}$

QUESTION 2

What is output by the code to the right?

- A. 330 B. 660 C. 1
D. 0.66 E. 0

```
double a = 0.0033;
double b = a * 10 * (10 * 2);
System.out.print( (int) b );
```

QUESTION 3

What is output by the code to the right?

- A. -12 B. -1 C. 0
D. 12 E. 13

```
int val = 0;
for(int i = 1; i < 13; i++)
    val--;
System.out.print(val);
```

QUESTION 4

What is output by the code to the right?

- A. $C\%E\&\&R-F$ B. $c\%e\&\&r-f$
C. $c5e77r-f$ D. $cerf$
E. $C\%e\&\&r-F$

```
String ts = "C%e&&r-F";
System.out.print(ts.toLowerCase());
```

QUESTION 5

What is output by the code to the right?

- A. 4A B. 5null C. 4C
D. 4n E. 4null

```
String[] lets = {"A", "C", null, "A", "D"};
System.out.print(lets.length);
System.out.print(lets[2]);
```

QUESTION 6

What is output by the code to the right?

- A. 0 B. 4 C. 6
D. 20 E. 22

```
int x2 = 2;
int y2 = 3;
int z2 = x2 + y2 * 10 * 100 / 3 / 100 * 2;
System.out.print(z2);
```

QUESTION 7

Which answer is logically equivalent to the following boolean expression, where w , x , y , and z are int variables?

$!((x \neq y) \ || \ (w \geq z))$

- A. $(x == y) \ \&\& \ (w < z)$ B. $(x \neq y) \ \&\& \ (w \geq z)$ C. $(x == y) \ || \ (w < z)$
D. $(x == y) \ || \ (w \leq z)$ E. $!(x \neq y) \ || \ !(w == z)$

<p>QUESTION 8</p> <p>What is output by the code to the right?</p> <p>A. 1 B. 14 C. 24 D. 34 E. 4</p>	<pre>int x3 = 14; int y3 = 14; int z3 = x3 / y3 * x3; if(x3 > y3 && x3 > z3) System.out.print(1); else if(y3 > z3 && y3 > x3) System.out.print(2); else if(z3 > y3 && z3 > x3) System.out.print(3); System.out.print(4);</pre>
<p>QUESTION 9</p> <p>What replaces <*1> in the code to the right so that the instance variable named <code>dir</code> in the <code>Critter</code> object being instantiated is set to the value stored in the parameter named <code>dir</code>?</p> <p>A. <code>Critter.dir = dir</code> B. <code>this.dir = dir</code> C. <code>int dir = dir</code> D. <code>dir = dir</code> E. <code>static int dir = dir</code></p>	<pre>public class Critter{ public static final int NORTH = 0; public static final int EAST = 1; public static final int SOUTH = 2; public static final int WEST = 3; private int dir; public Critter(int dir){ <*1>; } public int move(){ if(dir == EAST) dir = SOUTH; else dir = EAST; return dir; } } // client code Critter c1 = new Critter(); c1.move(); c1.move(); System.out.print(c1.dir);</pre>
<p>Assume <*1> is filled in correctly.</p>	
<p>QUESTION 10</p> <p>What is output by the client code to the right?</p> <p>A. -1 B. 0 C. 3 D. There is no output due to syntax errors in the client code. E. There is no output due to a runtime error caused by the client code.</p>	
<p>QUESTION 11</p> <p>What is output by the code to the right?</p> <p>A. 0 B. 1 C. 49 D. 73 E. 122</p>	<pre>int m = 73; int n = 49; System.out.print(m & n);</pre>
<p>QUESTION 12</p> <p>What is output by the code to the right?</p> <p>A. 3 B. 3.3 C. 3.33 D. -3 E. -3.33</p>	<pre>int m2 = 10 / -3; System.out.print(Math.abs(-m2));</pre>

<p>QUESTION 13</p> <p>What is output by the code to the right?</p> <p>A. CC++ B. C PHP C++ PHP</p> <p>C. CC++PHP D. C C++PHP</p> <p>E. CBPHP</p>	<pre>System.out.print("C"); System.out.print("C++"); System.out.print("\nPHP");</pre>
<p>QUESTION 14</p> <p>What is output by the code to the right?</p> <p>A. +1638 B. +16384 C. 20000</p> <p>D. +16380 E. 1638</p>	<pre>System.out.printf("%+4d", 16384);</pre>
<p>QUESTION 15</p> <p>What is returned by the method call <code>run("ABA")</code>?</p> <p>A. 3 B. 6 C. 9</p> <p>D. 12 E. 15</p>	<pre>public int run(String s){ int x = s.length(); s = s + s; int y = s.length(); return x + y; }</pre>
<p>QUESTION 16</p> <p>What is output by the code to the right?</p> <p>A. 1 B. 11 C. 18</p> <p>D. 28 E. 40</p>	<pre>String stars = ""; for(int i = 0; i < 7; i++) for(int j = 0; j < 4; j++) stars = stars + "*"; System.out.print(stars.length());</pre>
<p>QUESTION 17</p> <p>What is output by the code to the right?</p> <p>A. 4 B. 5 C. 6</p> <p>D. 7 E. 13</p>	<pre>String raw = "%xx*xx\$a++u^^"; String[] dats = raw.split("\\W+"); System.out.print(dats.length);</pre>
<p>QUESTION 18</p> <p>Which of the following can replace <*1> in the code to the right so that the output is 5?</p> <p>I. <code>name.indexOf("R") == 1</code> II. <code>name.length() == 14</code> III. <code>name != null</code></p> <p>A. I only B. II only C. III only</p> <p>D. I and II only E. II and III only</p>	<pre>String name = "Richard_M._Karp"; int x = (<*1>) ? 5 : 0; System.out.print(x);</pre>

<p>QUESTION 19</p> <p>What is output by the code to the right?</p> <p>A. 22 B. 21 C. 20</p> <p>D. 1 E. 0</p>	<pre>int vv = 0; for(int i = 0; i >= 20; i++) vv++; System.out.print(vv);</pre>												
<p>QUESTION 20</p> <p>Method <code>_AWFUL</code> to the right will not compile due to a single syntax error. Which of the following best describes the syntax error in the method?</p> <p>A. <code>_AWFUL</code> is not a valid method name.</p> <p>B. <code>x * 5</code> is not a valid length for an array.</p> <p>C. <code>for(;;)</code> is not a valid for loop.</p> <p>D. <code>short</code> is not a valid variable name.</p> <p>E. The <code>if</code> statement requires braces.</p>	<pre>public boolean[] _AWFUL(int x, String s) { boolean flags[] = new boolean[x * 5]; int short = 0; for(;;) { flags[short] = s.charAt(short++) > 50; if(short >= s.length()) break; } return flags; }</pre>												
<p>QUESTION 21</p> <p>What is output by the code to the right?</p> <p>A. 1 B. 1234 C. 123455</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 k = "1234.55"; Scanner sc = new Scanner(k); System.out.print(sc.nextInt());</pre>												
<p>QUESTION 22</p> <p>Which of the following is not a class in the Java Class Library?</p> <p>A. Structure B. String C. List D. Double E. Map</p>													
<p>QUESTION 23</p> <p>What is the largest value that can be output by method <code>showVal</code> to the right?</p> <p>A. 127 B. 128 C. 255</p> <p>D. 256 E. 2147483647</p>	<pre>public void showVal(byte b) { System.out.print(b); }</pre>												
<p>QUESTION 24</p> <p>Consider method <code>logic</code> to the right. When will the following conditions be true at the point marked <code>// question 24</code>?</p> <table border="1" data-bbox="162 1617 730 1890"> <tr> <td style="padding: 5px;"><code>x == 20</code></td> <td style="padding: 5px;"><code>y >= 0</code></td> </tr> <tr> <td style="padding: 5px;">A. never</td> <td style="padding: 5px;">never</td> </tr> <tr> <td style="padding: 5px;">B. never</td> <td style="padding: 5px;">sometimes</td> </tr> <tr> <td style="padding: 5px;">C. sometimes</td> <td style="padding: 5px;">sometimes</td> </tr> <tr> <td style="padding: 5px;">D. sometimes</td> <td style="padding: 5px;">always</td> </tr> <tr> <td style="padding: 5px;">E. always</td> <td style="padding: 5px;">always</td> </tr> </table>	<code>x == 20</code>	<code>y >= 0</code>	A. never	never	B. never	sometimes	C. sometimes	sometimes	D. sometimes	always	E. always	always	<pre>public int logic(int x, int y) { if(x != 5 x != 7) x = 20; else x *= 2; if(-100 < y && y < 100) y = y * y; else y = 1234; // question 24 return x * y; }</pre>
<code>x == 20</code>	<code>y >= 0</code>												
A. never	never												
B. never	sometimes												
C. sometimes	sometimes												
D. sometimes	always												
E. always	always												

QUESTION 25

Method `search` to the right is a flawed implementation of the binary search algorithm. The method does not always work as intended. Which of the following changes must be made so that the method correctly implements the binary search algorithm?

- A. change `while(r < 0 && s < e)`
to `while(r < 0 && s <= e)`
- B. change `int m = (s + e) / 2`
to `int m = s + e / 2`
- C. change `int e = data.length - 1`
to `int e = data.length + 1`
- D. change `return r`
to `return s`
- E. change `if(data[m] == t)`
to `if(data[m] == r)`

Assume method `search` has been corrected.

QUESTION 26

Which of the following is required as a pre condition so that method `search` works as intended and does not cause a runtime error? Pre conditions are the things that must be true before a method is called for the method to work.

- A. `data != null`
- B. `data != null` and the elements in data are ascending order
- C. `data != null` and `data.length > 0`
- D. `data != null` and `t != 0`
- E. `data != null` and `t != -1`

QUESTION 27

What is output by the code to the right?

- A. `r` B. `o` C. `114`
- D. `s` E. `R`

```

/* pre: question 26
   post: return an index of data such that
        data[return value] == t or -1 if t is
        not present in data.
*/
public int search(int[] data, int t) {
    int s = 0;
    int e = data.length - 1;
    int r = -1;
    while(r < 0 && s < e) {
        int m = (s + e) / 2;
        if( data[m] == t )
            r = m;
        else if(t > data[m])
            s = m + 1;
        else
            e = m - 1;
    }
    return r;
}

```

```

String winner = "AlanEmerson";
System.out.print(winner.charAt(8));

```

QUESTION 28

What is output by the code to the right?

- A. `4` B. `-4.0` C. `-3.0`
- D. `0` E. `3`

```

double neg = -1.7;
neg *= 2;
System.out.print(Math.ceil(neg));

```

QUESTION 29

What replaces **<*1>** in the code to the right so that the variable `it` refers to an `Iterator` object for the `ArrayList v`?

- A. `new Iterator<Integer>()`
- B. `new Iterator<Integer>(v)`
- C. `v.iterator<Integer>()`
- D. `v.iterator()`
- E. `new Iterator()`

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

QUESTION 30

What is output by the code to the right?

- A. 1210 B. 57 C. 512710
- D. 107125 E. 5710

QUESTION 31

Which of the following can replace **<*1>** in the code to the right so that the value stored in `p` is doubled?

- I. `p *= 2`
- II. `p = p * p`
- III. `p = p * 2`
- A. I only B. II only C. III only
- D. I and III only E. I, II, and III

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

QUESTION 32

What sorting algorithm does method `sort` implement?

- A. heap sort
- B. quick sort
- C. selection sort
- D. radix sort
- E. insertion sort

```
int[] data = {5, 12, 7, 10};
List<Integer> v = new ArrayList<Integer>();
for(int x : data)
    v.add(x);

Iterator<Integer> it = <*1>;
while(it.hasNext()) {
    int temp = it.next();
    if(temp < 10) {
        System.out.print(temp);
        it.remove();
    }
}
```

```
// pre: all elements in vals >= 0
public void sort(int[] vals) {
    int[] w = new int[vals.length];
    int[] h = new int[vals.length];
    int p = 1;
    for(int i = 0; i < 31; i++) {
        int w_i = 0, h_i = 0;
        for(int j = 0; j < vals.length; j++) {
            int d = vals[j] & p;
            if(d == 0)
                w[w_i++] = vals[j];
            else
                h[h_i++] = vals[j];
        }
        int i_o = 0;
        for(int j = 0; j < w_i; j++)
            vals[i_o++] = w[j];
        for(int j = 0; j < h_i; j++)
            vals[i_o++] = h[j];
        <*1>;
    }
}
```

QUESTION 33

What is the order of method `addAll` to the right if `s` is the following type of `Set`? `vals` contains N distinct values. Pick the most restrictive, correct set of values.

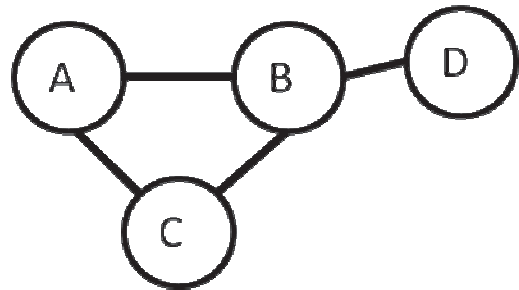
- | | | |
|----|---------------|---------------|
| | HashSet | TreeSet |
| A. | $O(N)$ | $O(N)$ |
| B. | $O(N)$ | $O(N \log N)$ |
| C. | $O(1)$ | $O(\log N)$ |
| D. | $O(N)$ | $O(N^2)$ |
| E. | $O(N \log N)$ | $O(N^2)$ |

```
// pre: s.size() == 0
public void addAll(Set<Integer> s,
                  int[] vals) {
    for(int elem : vals)
        s.add(elem);
}
```

QUESTION 34

What kind of graph does the picture to the right represent?

- A. a directed unweighted graph
- B. a directed weighted graph
- C. an undirected unweighted graph
- D. an undirected weighted graph
- E. a binary search tree

**QUESTION 35**

What replaces `<*1>` in method `isVowel` to the right so that the method returns true if the `char ch` is equal to 'a', 'e', 'i', 'o', or 'u'?

- A. `"aeiou".indexOf(ch) != -1`
- B. `ch == 'a' || 'e' || 'i' || 'o' || 'u'`
- C. `ch == "aeiou"`
- D. `ch |= "aeiou"`
- E. `ch.equals("aeiou")`

```
public boolean isVowel(char ch) {
    return <*1>;
}

public void sample() {
    String st = "abracadadra";
    int track = 0;
    for(int i = 0; i < st.length(); i++) {
        if(isVowel(st.charAt(i)))
            continue;
        track += i;
        i += 2;
    }
    System.out.print(track);
}
```

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

QUESTION 36

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

- A. 0
- B. 3
- C. 13
- D. 25
- E. 30

QUESTION 37

What is output by the code to the right?

- A. 2600
- B. 2500
- C. 260
- D. 27
- E. 26

```
Set<Character> vals;
vals = new HashSet<Character>();
for(int i = 0; i < 26; i++)
    for(int j = 0; j < 10; j++)
        vals.add( (char) (i + 'a') );
System.out.print(vals.size());
```


QUESTION 38

Given the class `Elem` to the right, what is output by the following client code?

```
Elem e1 = new Elem("X", null);
Elem e2 = new Elem("Y", e1);
Elem e3 = new Elem("Z", e2);
e1.n = e3;
Elem t = e2;
for(int i = 0; i < 32; i++)
    t = t.n;
System.out.print(t.d);
```

- A. e1
- B. e2
- C. X
- D. Y
- E. Z

```
public class Elem {
    public Elem n;
    public Object d;

    public Elem(Object dd, Elem nn) {
        n = nn;
        d = dd;
    }
}

public class Structure {

    private Elem f = new Elem(null, null);
    private int s = 0;
```

QUESTION 39

Given the classes `Elem` and `Structure` to the right, what is output by the following client code?

```
Structure st = new Structure();
st.add(37, 0);
st.add(42, 0);
st.add(13, 1);
st.add(17, 0);
System.out.print(st.size() + " " +
    st.get(2));
```

- A. 2 42
- B. 4 17
- C. 4 13
- D. 4 37
- E. 4 42

```
public void add(Object obj, int pos) {
    Elem t = getTo(pos - 1);
    t.n = new Elem(obj, t.n);
    s++;
}

public void remove(int pos) {
    Elem t = getTo(pos - 1);
    t.n = t.n.n;
    s--;
}

public Object get(int pos) {
    return getTo(pos).d;
}

public int size() {
    return s;
}

public Elem getTo(int pos) {
    Elem t = f;
    for(int i = 0; i < pos; i++, t = t.n);
    return t;
}
}
```

QUESTION 40

What type of data structure do the `Structure` and `Elem` classes implement?

- A. an array based list
- B. a heap
- C. a queue
- D. a binary search tree
- E. a linked list

No Test Material on This Page

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 (end - 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 str. 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 str, 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()
- o E get(int index)
- o E set(int index, E e)
Replaces the element at index with the object e.
- o void add(int index, E e)
Inserts the object e 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.ArrayList<E> implements List<E>

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.lang.Exception

```

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)

```