University Interscholastic League

Computer Science Competition

Number 139 (District 1 - 2013)

General Directions:

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 $\mathrm{A39}_{16}$ when this value is converted to base 8?

A. 2624_8 B. 2617_8 C. 5071_8 D. 5100_8 E. 5107_8

QUESTION 2

What is output by the code to the right?

A. 2 B. 2.0 C. 42

D. 12 E. There is no output.

```
int x = 42;
int y = 5;
int z = 3;
System.out.print(x/y/z);
```

QUESTION 3

What is output by the code to the right?

A. 2187 B. 729 C. 33

D. 315 E. 0

```
int m = 3;
for(int i = 0; i < 8; i += 2) {
   m += m*i;
}
System.out.print(m);
```

QUESTION 4

What output by the code to the right?

A. 6 B. 7 C. 8

D. 11 E. There is no output due to a runtime error.

```
String s = "JohnSteinbeck";
String t = s.substring(2, s.length()-4);
System.out.print(t.length());
```

QUESTION 5

What replaces <*1> in the code to the right so that the output is 5?

A. {0, 0, 0, 0, 0}

B. new r[5]

C. new [5]

D. new double[5]

E. Both A and D

```
double[] r = <*1>;
System.out.print(r.length);
```

QUESTION 6

What is output by the code to the right?

A. 0 B. 1

C. -1 D. -0.5

E. 11

```
double x = 56/4%5-2.5;
int y = (int) x;
System.out.print(y);
```

QUESTION 7

How many combinations of truth values for p, q and r make this expression to the right true?

A. 1 B. 4

C. 15 D. 7

E. 8

```
(p && !q) || (q || !r)
```

QUESTION 8

What is output by the code to the right?

A. 10 B. 4050 C. 5

D. 50 E. 110

```
String s = "";
int x = 50;
if(x % 2 == 0) {
   if(x % 3 == 0) {
      x /= 5;
   }
   else {
      s += x-10;
   }
}
else {
   s += 1;
   x %= 5;
}
System.out.print(s + x);
```

QUESTION 9

What is the value of the Java expression to the right?

A. 2 B. 8 C. 35

D. 43 E. 55

```
43^25&14
```

QUESTION 10

For int variables x and y, when is the boolean expression true?

I. x and y are both positive.

II. x and y are both even.

III. x and y are both odd.

IV. One of x and y is even and the other is odd.

A. I only

B. II only

C. III only

D. IV only

E. II and III

```
x%2 != y%2
```

QUESTION 11

What is $3B6_{16} + 54C_{16}$?

A. 800_{16} B. 900_{16} C. 902_{16} D. 968_{16} E. $A02_{16}$

QUESTION 12

Assume `int x` has been initialized to a positive even integer, and that `int y` has been initialized to a positive odd integer. What is the value of `k` after the code to the right is executed?

A. `0` B. `2x + y` C. `x + y`

D. `x * y` E. `(x*y)/2`

```
int k = 0;
for(int i = 0; i < x; i += 2) {
   for(int j = 0; j < y; j++) {
      k++;
   }
}
```

QUESTION 13

What is output by the code to the right?

A. `0` B. `6` C. `9`

D. `12` E. `25`

```
int y = 0;
int x = 641275;
while(x > 0) {
   switch(x%10) {
      case 0: case 2: case 4: case 6:
      case 8:
         y++;
         break;
      case 1: case 3: case 5: case 7:
      case 9:
         y += 2;
         break;
   }
   x = x/10;
}
System.out.print(y);
```

QUESTION 14

What code can replace **<*1>** so that instance variable `x` and instance variable `y` are both assigned the value `1`?

I. `x = 1; y = 1;`

II. `this();`

III. `this.x = 1; this.y = 1;`

A. I only B. II only

C. III only D. I, II and III

E. II and III

Assume that **<*1>** was replaced correctly.

QUESTION 15

Assume that the client code to the right is in some class other than the `Example` class. What code can replace **<*2>** so that the output from the client code is `2 1`?

A. None - no code is needed to replace **<*2>**.

B. `change(2);`

C. `e.change(2);`

D. `e.x = 2;`

E. `x = 2;`

```
public class Example {
   private int x;
   private int y;

   public Example() {
      x = 1;
      y = 1;
   }

   public Example(int x, int y) {
      <*1>
   }

   public void change(int n) {
      x = n;
   }

   public String toString() {
      return x + " " + y;
   }
}

// Client code in another class
Example e = new Example(2, 1);
<*2>
System.out.print(e);
```

QUESTION 16

What value is printed by the client code below?

```
int[] a = {6, 2, 4, 8, 1, 5};
System.out.print(mu(a));
```

A. 0 B. 5 C. 6

D. 9 E. 11

QUESTION 17

Let `n = r.length`. what is the run time of method `mu`? Give the most restrictive correct answer.

A. `O(1)` B. `O(logn)` C. `O(n)`

D. `O(nlogn)` E. O(n^2)

```
// pre: r != null and r.length >= 1
public static int mu(int[] r) {
   int pos = r.length - 1;
   int k = r[pos];
   int sum = 0;
   while(pos > 0) {
      sum += k;
      pos = pos/2;
      k = r[pos];
   }
   return sum;
}
```

QUESTION 18

What is output by the code to the right?

A. 0002 B. 00-12 C. 0001

D. 1101 E. 11-12

```
int x = 1;
int y = 1;
int z = --x - 1;
int t = (y--) + 1;
System.out.print("" + x + y + z + t);
```

GO ON TO THE NEXT PAGE.

QUESTION 19

To produce the correct output, which expression should replace **<*1>** in the code to the right? You will need to replace both <*1> and <*2> to get the desired output.

A. `j < i` B. `j < 7`

C. `j < 7-i` D. `j < 6-i`

E. `j < 6`

QUESTION 20

To produce the correct output, which expression should replace **<*2>** in the code to the right?

A. `j < i` B. `j < 7`

C. `j < 7-i` D. `j < 6-i`

E. `j < 6`

```
for(int i = 1; i < 7; i++) {
   for(int j = 1; <*1>; j++) {
      System.out.print("0");
   }
   for(int j = 0; <*2>; j++) {
      System.out.print("1");
   }
   System.out.println();
}

Desired Output from above code:
000001
000011
000111
001111
011111
111111
```

QUESTION 21

What is output by the code to the right? In the `Strings` below, ~~b~~ represents a blank space.

A. ~~bb~~Ford~~bbb~~13

B. Ford~~bbbbb~~13

C. ~~bb~~Ford00013

D. Ford~~bb~~00013

E. ~~bb~~Ford13~~bbb~~

```
System.out.printf("%-6s%05d", "Ford", 13);
```

QUESTION 22

What is output by the code to the right?

A. `2hello` B. `3bye` C. `0null`

D. `4hello` E. `4bye`

```
String[] arr = {"hello", "hi", "bye",
"hi"};
TreeSet<String> s = new TreeSet<String>();
for(String i : arr)
   s.add(i);
s.remove("hi");
System.out.print(s.size());
System.out.print(s.last());
```

QUESTION 23

What is output by the code to the right?

A. `atJa` B. `etJe` C. `PatJa`

D. `etJef` E. `atJaf`

```
String s = "Jefferson";
String t;
t = "Pat".concat(s).replace('a', 'e');
t = t.substring(1, 5);
System.out.print(t);
```

QUESTION 24

What is output by the code to the right?

A. `truetrue`

B. `truefalse`

C. `falsetrue`

D. `falsefalse`

E. There is no output due to a compilation error.

```
Integer y = new Integer(13);
int x = y;
String s = "13";
String r = ""+x;
int z = Integer.parseInt(s);
System.out.print(""+s.equals(r) + (y==z));
```

QUESTION 25

What is the value of the expression to the right?

A. `"12buckle my shoe34"`

B. `"3buckle my shoe7"`

C. `"12buckle my shoe7"`

D. `"3buckle my shoe34"`

E. This is not a valid Java expression.

```
1 + 2 + "buckle my shoe" + 3 + 4
```

QUESTION 26

What is returned by the method call `rec("abcdef")`?

A. `"f"` B. `"fdb"` C. `"fcba"`

D. `"eca"` E. `"fca"`

```
public static String rec(String s) {
   if(s == null || s.length() == 0)
      return "";
   return s.charAt(s.length()-1) +
rec(s.substring(0, s.length()/2));
}
```

GO ON TO THE NEXT PAGE.

QUESTION 27

The client code to the right appears in some class other than `Sub` or `Sup`. Which choice for **<*1>** will produce the output `Batman 3` from the statement labeled `line 1`?

```
I.   s = t;
II.  sup(t);
III. super(t);
```

A. I only B. II only C. III only

D. I and III E. I and II

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

QUESTION 28

Which choice for **<*2>** will produce the output `Ironman 1` from the statement labeled `line 2`?

```
I.   s = "Ironman";
II.  change("Ironman");
III. super.change("Ironman");
```

A. I only B. II only C. III only

D. I and III E. II and III

```
public class Sup {
   private String s;

   public Sup() {
      s = "Superman";
   }
   public Sup(String t) {
      s = t;
   }
   public void change(String t) {
      s = t;
   }
   public String getS(){
      return s;
   }
}

public class Sub extends Sup {
   private int cat;

   public Sub(String t, int c) {
      <*1>
      cat = c;
   }
   public void change(String t) {
      <*2>
      cat = 1;
   }
   public int getCat() {
      return cat;
   }
}

// Client code
Sub x = new Sub("Batman", 3);
System.out.print(x.getS() + " " +
x.getCat()); // line 1
x.change("Ironman");
System.out.print(x.getS() + " " +
x.getCat()); // line 2
```

QUESTION 29

Which of the following creates an `int` array `arr` that contains three zeros?

```
I.   int[] arr = new int[3];
II.  int[] arr = {0, 0, 0};
III. int[] arr = new 3[0];
```

A. I only B. II only C. III only D. I and II E. I, II and III

QUESTION 30

What is the value of k after the code to the right has been executed?

A. 10 B. 45 C. 50

D. 90 E. 1000

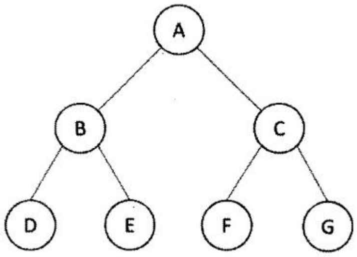
```
int k = 0;
for(int i = 0; i < 10; i++) {
   for(int j = 0; j < i; j++)
      k++;
}
```

QUESTION 31

Which of the following lists the nodes of the tree in the order that they would be visited in a preorder traversal?

A. ABDECFG B. DBEAFCG C. DEBFGCA

D. DEFGBCA E. ABCDEFG



QUESTION 32

What is `arr` after the code to the right is executed?

A. `{"Hello", "hi", "Bye"}`
B. `{"Hello-", "hi?", "Bye"}`
C. `{"Hello", "-hi", "?Bye"}`
D. `{"Hello", "hi"}`
E. `null`

```
String s = "Hello-hi?Bye";
String[] arr = s.split("\\?|-");
```

QUESTION 33

Which of the following can replace **<*1>** in the code to the right so that the method returns the sum of the last entries in each row of matrix `r`? For example, for

```
int[][] a = {{1, 2, 3}, {4, 5}, {6}};
```

the call `mat(a)` should return `14`.

I. `r[i].length-1`
II. `r[0].length-1`
III. `r[].length-1`

A. I only B. II only C. III only

D. I and II E. I, II and III

The parameter `r` can be a jagged array for questions 33 and 34.

```
public static int mat(int[][] r) {
   int s = 0;
   for(int i = 0; i < r.length; i++){
      s += r[i][<*1>]; // line 1
   }
   return s;
}
```

QUESTION 34

Suppose you want to modify the `mat` method so that it returns the sum of all elements in `r` that have an odd column index. Which of the following could be the body of the loop in the code to the right, replacing `line 1`?

I.
```
for(int j = 0; j < r[0].length;
j++)
  if(j % 2 == 1) s += r[i][j];
```

II.
```
for(int j = 1; j < r[0].length; j
+= 2)
  s += r[i][j];
```

III.
```
for(int j = 1; j < r[i].length; j
+= 2)
  s += r[i][j];
```

A. I only B. II only C. III only

D. I and II E. I, II and III

QUESTION 35

What is output by the code to the right?

A. `2 2` B. `1 2` C. `1 1`

D. `2 3` E. `1 3`

```
Stack<Integer> st = new Stack<Integer>();
st.push(1);
st.push(2);
st.push(3);
int i = st.search(2);
System.out.print(i);
System.out.println(" " + st.pop());
```

QUESTION 36

On an `int` array of length `n`, the best case run time of insertion sort is which of the following? Choose the most restrictive correct answer.

A. `O(1)`

B. `O(logn)`

C. `O((logn)`^{2}`)`

D. `O(n)`

E. `O(n`^{2}`)`

QUESTION 37

What is output by the code to the right?

A.
```
\
\a//
```

B.
```
\n
\a//
```

C.
```
\\n
\\a/
```

D. `\\n\n\\a/`

E. `\\n\n\\a//`

```
System.out.print("\\n\n\\a//");
```

QUESTION 38

This program (contained in file Q38.java) is compiled and then executed from a command line prompt as follows:

```
java Q38 2 elephants fly
```

What is the output?

A. `1`

B. `3`

C. `13`

D. `15`

E. There is no output due to a run time error.

```
public class Q38 {
  public static void main(String[] args) {
    int j = 0;
    for(int i = 0; i < args.length; i++)
      j += args[i].length();
    System.out.print(j);
  }
}
```

QUESTION 39

What value is returned by the call `tau("1234")`?

A. 34224

B. 34124

C. 343424

D. 234124

E. 234224

```
public String tau(String s) {
   String a = s.substring(s.length()/2);
   if(s.length() <= 2)
      return a + s.charAt(0);
   return a + tau(s.substring(1,
s.length()-2)) + s.charAt(s.length()-1);
}
```

QUESTION 40

What is output by the code to the right?

A. 53687 B. 47535

C. 36424 D. 23451

E. 37654

```
int[] a = {0, 1, 2, 3, 4, 5};
for(int i = 0; i < 4; i++)
   a[i] = a[3*a[i+1]%5] + 2;
a[4] = a[3*a[0]%5] + 2;
for(int i = 0; i < 5; i++)
   System.out.print(a[i]);
```

No Test Material on This Page

Standard Classes and Interfaces — Supplemental Reference

`class java.lang.Object`

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

`interface java.lang.Comparable<T>`

- `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>`

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

`class java.lang.Double implements Comparable<Double>`

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

`class java.lang.String implements Comparable<String>`

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

`class java.lang.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 java.lang.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, in b)`
- `static int max(int a, int b)`
- `static long round(double a)`
- `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>`

- `boolean add(E e)`
- `int size()`
- `Iterator<E> iterator()`
- `ListIterator<E> listIterator()`
- `E get(int index)`
- `E set(int index, E e)`
  Replaces the element at `index` with the object `e`.
- `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`.
- `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:

- `void addFirst(E e)`
- `void addLast(E e)`
- `E getFirst()`
- `E getLast()`
- `E removeFirst()`
- `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 District 1-2013

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

Notes:

Questions 17 and 36: 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.