University Interscholastic League

Computer Science Competition

Number 131 (Invitational A - 2012)

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 does ABC_{16} minus $1FF_{16}$ equal?

A. 853_{16} B. 953_{16} C. $8BD_{16}$ D. $8CD_{16}$ E. $9CD_{16}$

QUESTION 2

What is output by the code to the right?

A. 4 B. 6.67 C. 15

D. 15.3333 E. 24

```
int x = 3;
int y = 10 / x + x * 4;
System.out.print(y);
```

QUESTION 3

What is output by the code to the right?

A. 5 B. 12 C. 13

D. 25 E. 50

```
int val = 0;
int limit = 25;
for(int i = 1; i < limit; i += 2)
  val++;
System.out.print(val);
```

QUESTION 4

What is output by the code to the right?

A. BoBo2 B. BoBoBo C. BoBob

D. BoBoB E. 'BoBo2'

```
String name = "Bo";
name = name + name + 2;
System.out.print(name);
```

QUESTION 5

What is output by the code to the right?

A. 0 B. 0.0

C. 1.0 D. 4

E. The output will vary from one execution of the code to the next.

```
double[] list = new double[6];
System.out.print(list[4]);
```

QUESTION 6

What is output by the code to the right?

A. 1 B. 2 C. 8

D. 12 E. 16

```
int x1 = 2;
int y1 = x1 * x1 * x1 * x1;
System.out.print(y1);
```

QUESTION 7

What is output by the code to the right?

A. false false B. true false

C. false true D. true true

E. 0 1

```
boolean p = true, q = false;
System.out.print(p || q );
System.out.print(" " + (p && q));
```

QUESTION 8

What is output by the code to the right?

A. 24 B. 23 C. 13

D. 14 E. 1

```
String n2 = "126547";
if(n2.indexOf('a') != -1)
   System.out.print(1);
else
   System.out.print(2);
if(n2.length() > 6)
  System.out.print(3);
else
  System.out.print(4);
```

QUESTION 9

What replaces **<*1>** in the code to the right so that the output of the client code to the right is `go Longhorns`?

A. `mascot`

B. `School.mascot`

C. `String mascot`

D. `toString()`

E. `this.mascot`

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

QUESTION 10

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

```
School sc2 = new School();
System.out.print(sc2);
```

A. `go null`

B. `go`

C. `"go mascot"`

D. There is no output due to a syntax error.

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

```
public class School {
  private String mascot;

  public School(String mascot) {
    <*1> = mascot;
  }

  public String toString() {
    return "go " + mascot;
  }
}

// client code
School sc = new School("Longhorns");
System.out.print(sc.toString());
```

QUESTION 11

What is output by the code to the right?

A. true B. false C. 6

D. 111 E. 117

```
int m = 47;
int n = 70;
System.out.print(m | n);
```

QUESTION 12

What is the largest value that can be output by the code to the right?

A. 50 B. 55 C. 100

D. 110 E. 150

```
int tot = 0;
for(int i = 0; i < 10; i++) {
  int temp = (int) (Math.random() * 11);
  tot += temp - 5;
}
System.out.print(tot);
```

QUESTION 13

What is output by the code to the right?

A.
```
cat
dog
ape
```

B.
```
cat dog
ape
```

C.
```
cat
dog ape
```

D.
```
apecatdog
```

E.
```
catdogape
```

```
System.out.print("cat");
System.out.print("dog");
System.out.println("ape");
```

QUESTION 14

What is output by the code to the right?

A. `a4` B. `14.26730` C. `14.2673`

D. `14.2672` E. `14.267299`

```
double a4 = 14.267299;
System.out.printf("%7.4f", a4);
```

QUESTION 15

What is returned by the method call `eval(5, 3)`?

A. `12` B. `32` C. `36`

D. `40` E. `4096`

```
public int eval(int y, int x) {
  y += x;
  x++;
  return y * x;
}
```

QUESTION 16

What is output by the code to the right?

A. `20` B. `22`

C. `45` D. `200`

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

```
String stars = "";
for(int i = 0; i < 10; i++)
  stars += "*";
for(int i = 0; i < 10; i++)
  stars += "*";
System.out.println(stars.length());
```

QUESTION 17

Which of the following Java expressions is equivalent to the formula to the right? `a`, `b`, and `c` are variables of type `double`.

A. `(-b + Math.sqrt(b ^^ 2 - 4 * a * c)) / (2 * a)`

B. `(-b +- Math.sqrt(b ** 2 - 4 * a * c)) / (2 * a)`

C. `(-b + (b * b - 4 * a * c) ^ 0.5) / (2 * a)`

D. `-(b + Math.sqrt(b * b - 4 * a * c)) / 2a`

E. `(-b + Math.sqrt(b * b - 4 * a * c)) / (2 * a)`

$$\frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

QUESTION 18

What is output by the code to the right?

A. `10` B. `12` C. `14`

D. `15` E. `17`

```
String garbage = "1000\\100\"\t+5";
System.out.print(garbage.length());
```

QUESTION 19

What is output by the code to the right?

A. `5` B. `7` C. `35`

D. There is no output due to a syntax error.

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

```
final int rate;
int sts = 7;
int profit = 5;
rate = sts * profit;
System.out.print(rate);
```

QUESTION 20

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

```
(x != y) || !(y >= z)
```

A. `(x != y) && !(y >= z)` B. `(x == y) && (y == z)` C. `!((x == y) || (y >= z))`

D. `(x != z) && (y == z)` E. `!((x == y) && (y >= z))`

QUESTION 21

Method `Total` to the right will not compile due to a syntax error. Which of the following best explains the syntax error in method `Total`?

A. `Total` is not a legal method name.

B. The keyword `static` must be removed from the method header.

C. The variable `res` is not initialized.

D. A `char` may not be added to a variable of type `int`.

E. The `for` loop must have a set of braces, `{}`.

```
public static int Total(String st) {
  int res;
  for(int i = 0; i < st.length(); i++)
    res += st.charAt(i);
  return res;
}
```

QUESTION 22

What is output by the code to the right?

A. `0` B. `null`

C. The code runs, but there is no output.

D. There is no output due to a syntax error.

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

```
String[] names = new String[4];
System.out.print(names[2].length());
```

QUESTION 23

What is output by the code to the right?

A. `0` B. `1.0` C. `1`

D. `1.946465` E. `2`

```
double a5 = 3.89293;
a5 /= 2;
System.out.print( (int) a5 );
```

QUESTION 24

Which of the following could replace **`<*1>`** so that the following line of code compiles without syntax error?

```
int <*1> = 15;
```

A. `x5` B. `5_x` C. `_5` D. `x-5` E. More than one of A through D is correct.

QUESTION 25

What is output by the code to the right?

A. `[C, A, B]` B. `[A, B]`

C. `[A, A]` D. `[C, B, A]`

E. `['C', 'A', 'B']`

```
ArrayList<Character> grades;
grades = new ArrayList<Character>();
grades.add('A');
grades.add('A');
grades.add(1, 'B');
grades.add(0, 'C');
grades.remove(1);
System.out.println(grades);
```

QUESTION 26

Method `get` to the right contains a logic error. Which of the following will occur when the method call `get("aaaa")` is made?

A. The program will crash due to a `StackOverError`.

B. Nothing. An infinite loops occurs.

C. The program will crash due to a `NullPointerException`.

D. The program will crash due to an `OutOfMemoryError`.

E. The program will crash due to an `IndexOutOfBoundsException`.

```
public String get(String st) {
  String r = "";
  for(int i = 0; i < st.length(); i *= 2)
    r = r + st.charAt(i) + r;
  return r;
}
```

QUESTION 27

What is output by the code to the right?

A. `false true` B. `true false`

C. `false false` D. `true true`

E. `false true true`

```
int v1 = 15;
int v2 = 30;
System.out.print((v1 >= v2) + " ");
System.out.print(v1 > 0 && v2 % v1 == 0);
```

QUESTION 28

What replaces **<*1>** in method `check` to the right so that `diff` is incremented if the element at index `i` in `a` does not equal the element at index `i` in `b`?

A. `a[i] != b[i]`

B. `!(a[i].equals(b[i]))`

C. `a.get(i) != b.get(i)`

D. `a[i].compareTo(b[i]) != 0`

E. More than one of A through D is correct.

```
public int check(int[] a, int[] b) {
  int diff = 0;
  for(int i = 0; i < a.length; i++)
    if( <*1> )
      diff++;
  return diff;
}
```

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

QUESTION 29

What is output by the client code to the right?

A. `0` B. `2` C. `3`

D. `12` E. `14`

```
// client code
int[] h1 = {5, -2, 4, 10, 45};
int[] h2 = {5, 2, -4, 10, 45};
System.out.print(check(h1, h2));
```

QUESTION 30

What is output by the code to the right?

A. `10000000` B. `640`

C. `0.15625` D. `0`

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

```
int bw = 10;
bw = bw >> 6;
System.out.print(bw);
```

QUESTION 31

An array with 1,000,000 distinct `int`s in random order is passed to a method that uses the heapsort algorithm, it takes 4 seconds for the method to complete. What is the expected time for the method to complete when sorting an array with 4,000,000 distinct `int`s in random order?

A. 1 second B. 4 seconds C. 8 seconds D. 17.6 seconds E. 64 seconds

QUESTION 32

Which of the following replaces **<*1>** in the code to the right so that the body of the `if` statement is executed if the element at position `j - 1` in `vs` is greater than the element at position `j`?

A. `vs.get(j-1).compareTo(vs.get(j)) > 0`

B. `vs[j-1].compareTo(vs[j]) > 0`

C. `vs.get(j-1).compareTo(vs.get(j))`

D. `vs.get(j-1) <= => vs.get(j)`

E. None of answers A through D are correct.

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

QUESTION 33

Which of the following replaces **<*2>** in the code to swap the elements at positions `j - 1` and `j` in `vs`?

A. `vs.set(j, vs.get(j - 1))`

B. `vs.set(j - 1, vs.remove(j))`

C. `vs.set(j, vs.set(j-1, vs.get(j)))`

D. `vs.set(j - 1, vs.get(j + 1))`

E. None of answers A through D are correct.

Assume **<*2>** is filled in correctly.

QUESTION 34

Which sorting algorithm does method `sort` implement?

A. radix sort B. insertion sort

C. selection sort D. quick sort

E. None of answers A through D are correct.

```
public void sort(ArrayList<Integer> vs) {
  for(int i = 0; i < vs.size(); i++)
    for(int j = 1; j < vs.size(); j++)
      if( <*1> )
        <*2>;
}
```

QUESTION 35

What is returned by the method call `add("aaaa")`?

A. 32 B. 80 C. 164

D. 200 E. 228

```
public int add(String s) {
  if(s.length() > 20)
    return s.length();
  else
    return add(s + s) + add(s + s + s);
}
```

QUESTION 36

What is returned by method `handle` if `t` is the matrix shown below?

1	4	0	2	1	6
0	-1	5	4	0	-4
2	2	7	1	13	2
10	5	13	13	4	20
1	4	2	1	3	2
0	-6	-5	3	-4	5

A. -88 B. -42 C. -11

D. -10 E. 8

```
public int handle(int[][] t) {
  int res = 0;
  for(int i = 0; i < t.length; i++) {
    int t1, t2;
    t1 = t2 = 0;
    for(int j = i; j < t.length; j++) {
      t1 += t[i][j];
      t2 += t[j][i];
    }
    if(t1 == t2)
      res += t1;
    else
      res -= t2;
  }
  return res;
}
```

QUESTION 37

What is output by the code to the right?

A. 13 17 B. -10 -10 50 50 -10 -20

C. 17 13 D. 50 50

E. -20 -10 50 50 -10 -10

```
Stack<Integer> st = new Stack<Integer>();
int[] data = {13, 17, -20, 50, -10};
for(int i : data)
  if(i % 5 == 0) {
    st.push(i);
    st.push(i > 0 ? i : -10);
  }
while(!st.isEmpty())
  System.out.print(st.pop() + " ");
```

GO ON TO THE NEXT PAGE.

QUESTION 38

What is output by the following client code?

```
String sch = "texasutamtechstate";
Structure<Character> st1;
st1 = new Structure<Character>();
for(int i = 0; i < sch.length(); i++)
  st1.add(sch.charAt(i));
String temp1 = st1.toString();
String[] res;
res = temp1.split("[\\s,\\[\\]]+");
for(String s3 : res)
    System.out.print(s3);
```

A. `texasutamtechstate`

B. `texasumch`

C. `hcmusaxet`

D. The program runs without error, but there is no output.

E. The output will vary from one run of the program to the next.

QUESTION 39

Given a `Structure` that contains N `Integer`s what is the order (Big O) of the add method for a value that is not already present in the `Structure`? Pick the most restrictive correct answer.

A. O(1) B. O(N) C. O(NlogN)

D. O(logN) E. $O(N^2)$

QUESTION 40

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

A. a graph B. a set

C. a stack D. an array based list

E. a linked list

```
public class Structure<E> {

  private ArrayList<E> con;

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

  public void add(E obj) {
    if(!con.contains(obj))
      con.add(0, obj);
  }

  public boolean present(E obj) {
    return con.contains(obj);
  }

  public String toString() {
    return con.toString();
  }

  public boolean remove(E obj) {
    return con.remove(obj);
  }
}
```

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

interface java.util.Queue<E>
- boolean add(E e)
- boolean isEmpty()
- E peek()
- E remove()

class java.util.PriorityQueue<E>
- boolean add(E e)
- boolean isEmpty()
- E peek()
- E remove()

interface java.util.Set<E>
- boolean add(E e)
- boolean contains(Object obj)
- boolean remove(Object obj)
- int size()
- Iterator<E> iterator()
- boolean addAll(Collection<? extends E> c)
- boolean removeAll(Collection<?> c)
- 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>
- 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()

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

interface java.util.Iterator<E>
- boolean hasNext()
- E next()
- void remove()

interface java.util.ListIterator<E> extends java.util.Iterator<E>

Methods in addition to the `Iterator` methods:
- void add(E e)
- void set(E e)

class java.lang.Exception
- Exception()
- Exception(String message)

class java.util.Scanner
- Scanner(InputStream source)
- boolean hasNext()
- boolean hasNextInt()
- boolean hasNextDouble()
- String next()
- int nextInt()
- double nextDouble()
- String nextLine()
- Scanner useDelimiter(String pattern)

Computer Science Answer Key
UIL Invitational A 2012

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

Notes: 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.

10. With the addition of a non-default constructor, the built in default constructor is no longer available.

24. Choices A and C are both correct.

34. The sorting algorithm used is the bubble sort.

University Interscholastic League

Computer Science Competition

Number 132 (Invitational B - 2012)

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 does 101_2 times 111_2 equal?

A. -10_2 B. 1100_2 C. 101111_2 D. 111101_2 E. 100011_2

QUESTION 2

What is output by the code to the right?

A. 18 B. 93 C. 793

D. 869 E. 1017

```
int x = 1776;
int y = x % 1000 + x / 100;
System.out.print(y);
```

QUESTION 3

What is output by the code to the right?

A. 2 B. 14 C. 24

D. 30 E. 222222222222

```
int val = 0;
for(int i = -2; i <= 12; i++)
  val += 2;
System.out.print(val);
```

QUESTION 4

What is output by the code to the right?

A. rmian.Basi B. ermian.Bas

C. rmian D. rmian.

E. UT.Permian.Basin

```
String c1 = "UT.Permian.Basin";
String c2 = c1.substring(5, 10);
System.out.print(c2);
```

QUESTION 5

What is output by the code to the right?

A. 7 1 B. 6 1 C. 7 13

D. 7 4 E. 6 13

```
int[] st = {5, 3, 13, 4, -1, 6, 0};
System.out.print(st.length + " " + st[3]);
```

QUESTION 6

What is output by the code to the right?

A. 9 B. 10 C. 11

D. 12 E. 20

```
int x1 = 3;
int y1 = 2;
int z1 = x1++ * ++y1;
System.out.print(z1);
```

QUESTION 7

How many combinations of values for the `boolean` variables `p`, `q`, and `r` will result in `s` being set to `true`?

A. 7 B. 5 C. 4

D. 1 E. 0

```
boolean p, q, r;
//code to initialize p, q, and r

boolean s = !p || !q || !r;
```

QUESTION 8

What is output by the code to the right?

A. 11 B. 12 C. 1x2

D. 15 E. 25

```
int x2 = 5;
if(x2 % 2 == 0)
  System.out.print(1);
else
  System.out.print(2);
System.out.print(x2);
```

QUESTION 9

What is output when the statement in the client code to the right marked `// line 1` is executed?

A. 321

B. 123

C. 3

D. 1

E. 31

QUESTION 10

What is output by the statement in the client code to the right marked `// line 2` ?

A. numStudentsisPrivate

B. "101"

C. 101

D. true100

E. 100true

```
public class School {
  private boolean isPrivate;
  private int numStudents;

  public School() {
    this(true);
    System.out.print(1);
  }

  public School(boolean p) {
    this(100, p);
    System.out.print(2);
  }

  public School(int ns, boolean p) {
    isPrivate = p;
    numStudents = ns;
    System.out.print(3);
  }

  public String toString() {
    return "" + numStudents + isPrivate;
  }
}

// client code
School sc = new School(); // line 1
System.out.print(sc); // line 2
```

QUESTION 11

What is output by the code to the right?

A. 1 B. 3 C. 10

D. 15 E. 21

```
int m = 0xA;
int n = 31;
System.out.print(m ^ n);
```

QUESTION 12

What is output by the code to the right?

A. 15.0 B. 6.0 C. 5.5

D. 4.0 E. 0

```
double m1 = 30.0;
m1 = Math.max(Math.sqrt(m1), m1 / 2);
System.out.print(m1);
```

QUESTION 13

What is output by the code to the right?

A.
```
red\
blue\
pink\
```

B.
```
red
blue
pink
```

C.
```
red
bluepink
```

D.
```
redblue
pink
```

E.
```
redbluepink
```

```
System.out.print("red\nblue\npink");
```

QUESTION 14

What is output by the code to the right?

A. `6.0`

B. `6528221.0`

C. `652.1`

D. `6,528,221.00`

E. `6,528,221.0`

```
double mon = 6528221.00;
System.out.printf("%,3.1f", mon);
```

QUESTION 15

What is returned by the method call `b(5)`?

A. `21`

B. `24`

C. `30`

D. `36`

E. `42`

```
public int a(int x, int z) {
  x++;
  z *= 2;
  return x + z;
}

public int b(int y) {
   return y + a(y, y);
}
```

QUESTION 16

What is output by the code to the right?

A. `50`

B. `30`

C. `25`

D. `15`

E. `10`

```
String stars = "";
for(int i = 0; i < 5; i++)
  for(int j = i; j < 5; j++)
    stars += "**";
System.out.print(stars.length());
```

QUESTION 17

Method `check` to the right will not compile due to a syntax error. Which of the following best describes the syntax error(s) in method `check`?

A. The line `x += a % 10;` causes a loss of precision error.

B. `&` is not a valid `boolean` operator.

C. Variables may not be named `continue`.

D. `x < 1000.00` is not a valid `boolean` expression.

E. More than one of A through D is correct.

```
public int check(double a) {
  boolean continue = true;
  int x = 0;
  while(continue & a > 1.0) {
    x += a % 10;
    a /= 10;
    continue = x < 1000.00;
  }
  return x;
}
```

QUESTION 18

What is the smallest possible value that will be printed out by the code to the right?

A. `0` B. `10` C. `30`

D. `60` E. `70`

```
int total = 0;
for(int i = 0; i < 10; i++)
  total += (int)(Math.random() * 4) + 3;
System.out.print(total);
```

QUESTION 19

Which of the following can replace **<*1>** in the code to the right so that the code segment compiles without error?

A. `double` B. `float`

C. `int` D. `long`

E. More than one of A through D is correct.

```
int xVal = 45;
int yVal = 100 * xVal;
<*1> vel = xVal / yVal;
```

QUESTION 20

Which answer is logically equivalent to the following `boolean` expression, where `p` and `q` are `boolean` variables?

```
p ^ q
```

A. `(!p && !q)` B. `(p && !q) || (!p && q)` C. `(!p || !q)`

D. `(!p || p) && (!q || q)` E. `(p && q) && !(p || q)`

QUESTION 21

What replaces **<*1>** in the code to the right to handle all values of `gm` that are not explicitly handled by one of the `case` sections?

A. `goto` B. `default` C. `case`

D. `break` E. `switch`

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

QUESTION 22

What is returned by the method call `pts("WWSLTLDSS")`?

A. `1013` B. `48` C. `45`

D. `17` E. `15`

```
public int pts(String res) {
  int t = 0;
  for(int i = 0; i < res.length(); i++) {
    char gm = res.charAt(i);
    switch(gm) {
      case 'D': t += 1; break;
      case 'L': t -= 1; break;
      case 'S': t += 4; break;
      case 'T': t += 2; break;
      case 'W': t += 1; break;
      <*1> : t += 1000; break;
    }
  }
  return t;
}
```

QUESTION 23

What is output by the code to the right?

A. `null` B. `10`

C. `1` D. `0`

E. The output will vary from one run of the program to the next

```
ArrayList<String> names;
names = new ArrayList<String>();
System.out.print(names.size());
```

QUESTION 24

What is output by the code to the right?

A. 24 B. 21 C. 15

D. 0 E. -5

```
int[] scs = {-5, 5, 2, -2, -5, 5};
int temp = 0;
for(int i : scs)
  temp = i + temp;
System.out.print(temp);
```

QUESTION 25

Given an array of 1000 elements in sorted order what is the largest possible value that will be printed when the array is passed to method `mystery`?

A. 0 B. 1 C. 9

D. 10 E. 500

QUESTION 26

Which algorithm does method `mystery` implement?

A. insertion sort B. selection sort

C. linear search D. radix sort

E. binary search

```
public int mystery(int[] v, int t) {
  int w = 0;
  int h = v.length - 1;
  int c = 0;
  while(w <= h) {
    c++;
    int m = (w + h) >>> 1;
    if(v[m] < t) h = m - 1;
    else if(v[m] > t) w = m + 1;
    else {
      System.out.print(c);
      return m;
    }
  }
  System.out.print(c);
  return -(w + 1);
}
```

QUESTION 27

Consider the following timing data for method `sort` shown to the right and various arrays:

array W: 1,000,000 elements in random order. Method `sort` takes 10 second to complete.

array X: 1,000,000 elements in ascending order. Method `sort` takes 100 seconds to complete.

What is the expected time for method `sort` to complete given array Y with 2,000,000 elements in random order and array Z with 2,000,000 elements in ascending order?

	array Y	array Z
A.	10 seconds	100 seconds
B.	11 seconds	400 seconds
C.	20 seconds	200 seconds
D.	21 seconds	210 seconds
E.	21 seconds	400 seconds

QUESTION 28

Which sorting algorithm do methods `hp` and `sort` implement?

A. radix sort B. mergesort

C. heap sort D. quicksort

E. selection sort

```
public void hp(double[] v, int i, int j) {
  double t = v[i];
  v[i] = v[j];
  v[j] = t;
}

public void sort(double[] v, int s, int p){
  if(s < p) {
    int m = (s + p) / 2;
    hp(v, m, s);
    int i, j = s;
    for(i = s + 1; i <= p; i++ )
      if(v[i] <= v[s]) {
        j++;
        hp(v, i, j);
      }
    hp(v, s, j);
    sort(v, s, j - 1);
    sort(v, j + 1, p);
  }
}
```

QUESTION 29

Which of the following replaces **<*1>** in the code to the right to indicate the `TDPoint` class is a subclass of the `Point` class?

A. `final` B. `static` C. `extends`

D. `super` E. `implements`

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

QUESTION 30

What is output by the following client code?

```
Point p1 = new Point(5, 2);
Point p2 = new Point(5, 2);
System.out.print(p1 == p2);
System.out.print(" " + p1.equals(p2));
```

A. `false false`

B. `false true`

C. `true false`

D. `true true`

E. There is no output due to a syntax error in the client code.

QUESTION 31

What is output by the following client code?

```
TDPoint p3 = new TDPoint(2, 3, 4);
System.out.print(p3);
```

A. `2324` B. `235`

C. `11` D. `236`

E. There is no output due to a syntax error in the client code.

```
public class Point {
  private int x, y;

  public Point(int xn, int yn) {
    x = xn;
    y = yn;
  }

  public int dFact() { return x * y; }

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

public class TDPoint <*1> Point {
  private int z;

  public TDPoint(int xn, int yn, int zn) {
    super(xn, yn);
    z = zn;
  }

  public int dFact() {
    return super.dFact() * z;
  }
}
```

QUESTION 32

What is returned by the method call `tester(20)`?

A. `80` B. `40` C. `10`

D. `5` E. `2`

```
public int tester(int x) {
  try {
    if(x < 10) return x * 2;
    return 100 / x;
  }
  finally { x *= 2; }
}
```

QUESTION 33

What is output by the code to the right?

A. `-5.15` B. `0.0` C. `0.15`

D. There is no output due to a syntax error.

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

```
PriorityQueue<Double> pq;
pq = new PriorityQueue<Double>();
pq.add(0.15);
pq.add(-5.15);
pq.add(0.0);
System.out.print(pq.peek());
```

QUESTION 34

Given the undirected, weighted graph to the right, what is the cost of the lowest cost path from vertex A to vertex E?

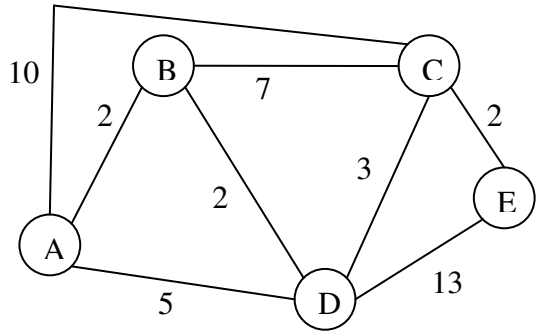
A. 1

B. 2

C. 9

D. 12

E. 14



QUESTION 35

What is output by the code to the right?

A. 0

B. 1

C. 25

D. 49

E. 81

```
int[][][] cb = {{{5, 6, 2}, {4, 3, 1}},
                {{2, 1, 4}, {7, -6, 1}}};
int m = 0;
for(int i = 0; i < cb[0].length; i++)
  for(int j = 0; j < cb[0][0].length; j++){
    int t = cb[0][i][j] - cb[1][i][j];
    t *= t;
    if(t > m)
      m = t;
  }
System.out.print(m);
```

QUESTION 36

What is output by the code to the right?

A. 0 B. 1 C. 4

D. 5 E. 6

```
String[] cs = {"ut", "ttu", "tamu", "tsu",
                                    "tu", "nt"};
int ct = 0;
for(String str : cs)
  if(str.matches("t.+"))
    ct++;
System.out.print(ct);
```

QUESTION 37

What is output by the client code to the right?

A. falsefalse

B. falsetrue

C. truefalse

D. truetrue

E. false is output and then a runtime error occurs.

```
public boolean check(int s, int b, int g) {
  return g <= b * 5 + s && g % 5 <= s;
}

// client code
System.out.print(check(3, 2, 8));
System.out.print(check(6, 0, 11));
```

GO ON TO THE NEXT PAGE.

QUESTION 38

What replaces **<*1>** in the `access` and `remove` methods to the right so that the methods generate an exception if the `boolean` expression `d == t` is `true`?

A. `throw` B. `try` C. `catch`

D. `double` E. `throws`

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

QUESTION 39

What is output by the following client code?

```
Structure gar = new Structure();
gar = gar.add("LHN");
gar = gar.add(24);
gar = gar.add('A');
while(!gar.isEmpty()) {
  System.out.print(gar.access());
  gar = gar.remove();
}
```

A. `LHN`

B. `LHNA24`

C. `A`

D. `A24LHN`

E. There is no output due to a syntax error in the client code.

QUESTION 40

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

A. a set

B. a queue

C. a binary search tree

D. a min heap

E. a stack

```
public class Structure {

  private static final Object t;
  static { t = new Object(); }

  private Object d;
  private Structure n;

  public Structure() { d = t; }

  public Structure add(Object d) {
    Structure r = new Structure();
    r.d = d;
    r.n = this;
    return r;
  }

  public Object access() {
    if(d == t)
      <*1> new IllegalStateException();
    return d;
  }

  public boolean isEmpty() {return d == t;}

  public Structure remove() {
    if(d == t)
      <*1> new IllegalStateException();
    return n;
  }
}
```

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

interface java.util.Queue<E>
- boolean add(E e)
- boolean isEmpty()
- E peek()
- E remove()

class java.util.PriorityQueue<E>
- boolean add(E e)
- boolean isEmpty()
- E peek()
- E remove()

interface java.util.Set<E>
- boolean add(E e)
- boolean contains(Object obj)
- boolean remove(Object obj)
- int size()
- Iterator<E> iterator()
- boolean addAll(Collection<? extends E> c)
- boolean removeAll(Collection<?> c)
- 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>
- 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()

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

interface java.util.Iterator<E>
- boolean hasNext()
- E next()
- void remove()

interface java.util.ListIterator<E> extends java.util.Iterator<E>

Methods in addition to the `Iterator` methods:
- void add(E e)
- void set(E e)

class java.lang.Exception
- Exception()
- Exception(String message)

class java.util.Scanner
- Scanner(InputStream source)
- boolean hasNext()
- boolean hasNextInt()
- boolean hasNextDouble()
- String next()
- int nextInt()
- double nextDouble()
- String nextLine()
- Scanner useDelimiter(String pattern)

Computer Science Answer Key
UIL Invitational B 2012

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

Notes:

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.

19. All of A through D are correct.

University Interscholastic League

Computer Science Competition

Number 133 (District 1 - 2012)

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 the sum of 757_8 and 540_8?

A. 1317_8 B. 1317_{10} C. 1297_{10} D. 217_8 E. 1517_8

QUESTION 2

What is output by the code to the right?

A. 1.0 B. 1.75 C. 2.15

D. 2.55 E. 2.75

```
double a = 3.5;
double b = 4 / 5 + a / 2;
System.out.print(b);
```

QUESTION 3

What is output by the code to the right?

A. 20 B. 40 C. 42

D. 60 E. 63

```
int val = 0;
for(int i = 0; i < 20; i++){
  val++;
  val += 2;
}
System.out.print(val);
```

QUESTION 4

What is output by the code to the right?

A. ibe B. v205 C. v_l

D. Be E. iBe

```
String c1 = "Tyler_Brownsville";
String c2 = "" + c1.charAt(12);
c2 += c1.charAt(5) + "" + c1.charAt(15);
System.out.print(c2);
```

QUESTION 5

What is output by the code to the right?

A. false false 6 B. false true 5

C. true false 5 D. true true 6

E. The output will vary from one run of the program to the next.

```
boolean[] st = new boolean[5];
System.out.print(st[3] + " " + !st[0]);
System.out.print(" " + st.length);
```

QUESTION 6

What is output by the code to the right?

A. -5.0 5.99 B. -10.0 5.99

C. -10 5.0 D. -5.0 5.0

E. 10.0 5.0

```
double a1 = 5.99;
double b1 = -1.25;
double c1 = ((int) a1) * ((int) b1);
System.out.print(c1 + " " + a1);
```

QUESTION 7

Which answer is logically equivalent to the following `boolean` expression, where `p` and `q` are `boolean` variables?

```
!(p && !q)
```

A. `!p || q` B. `!p && q` C. `!p && !q`

D. `p || q` E. `p || !q`

QUESTION 8

What is output by the code to the right?

A. 1 B. 2 C. 13

D. 23 E. 123

```
int x2 = 10;
if(x2 * x2 > x2 + x2)
  if(x2 > 0)
    System.out.print(1);
  else
    System.out.print(2);
System.out.print(3);
```

QUESTION 9

Which of the following can replace **<*1>** so that the class `College` to the right compiles without syntax error?

A. `super(s)` B. `this(s)`

C. `mascot = s` D. `School(s)`

E. More than one of A through D is correct.

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

QUESTION 10

What is output by the client code to the right?

A. `Raiders` B. `!!`

C. `!!Raiders` D. `Raiders!!`

E. There is no output due to a syntax error in the client code.

```
public class School {
  private String mascot;

  public School(String m) {mascot = m;}

  public void rah() {mascot += "!";}

  public String toString() {return mascot;}
}

public class College extends School {
  public College(String s) { <*1>; }
}

// client code
College tt = new College("Raiders");
tt.rah();
tt.rah();
System.out.print(tt);
```

QUESTION 11

What is output by the code to the right?

A. 282475249 B. 10000000 C. 15

D. 17 E. 13

```
int m = 7;
int n = 10;
System.out.print(m ^ n);
```

QUESTION 12

What is output by the code to the right?

A. 3125 B. 625 C. 625.0

D. 256 E. 125

```
int m1 = 4;
int n1 = (int) Math.pow(m1 + 1, m1 - 1);
System.out.print(n1);
```

QUESTION 13

What is output by the code to the right?

A. `1\23\4` B. `1 23 4`

C. `1t23t4` D. `1 2`

E. `1 2 3 4`

```
System.out.print("1\t2");
System.out.print("3\t4");
```

QUESTION 14

What is output by the code to the right?

A. `0045.70` B. `045.7` C. `45.70`

D. `+45.7` E. `045.70`

```
double tk = 45.7;
System.out.printf("%05.2f", tk);
```

QUESTION 15

What is returned by the method call `pick("universityTexas")` ?

A. `nvriyea` B. `unvi`

C. `niet` D. `unvis`

E. There is no output due to an infinite loop.

```
public String pick(String n) {
  String result = "";
  int lim = n.length();
  for(int i = 1; i <= lim; i *= 2)
    result += n.charAt(i);
  return result;
}
```

QUESTION 16

What is output by the code to the right?

A. `3` B. `10` C. `30`

D. `54` E. `75`

```
String stars = "";
for(int i = 0; i < 5; i++)
  for(int j = 0; j < 5; j++)
    stars += "***";
System.out.print(stars.length());
```

QUESTION 17

What is output by the line marked `// line 1` in the client code to the right?

A. `null` B. `200` C. `10 20`

D. There is no output due to a syntax error in the client code.

E. The output will vary from one run of the program to the next.

QUESTION 18

What is output by the line marked `// line 2` in the client code to the right?

A. `false false`

B. `false true`

C. There is no output due to a syntax error in the client code.

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

E. The output will vary from one run of the program to the next.

```
public class GasTank {
  private int cap;
  private int mpg;

  public GasTank(int c, int m) {
    cap = c;
    mpg = m;
  }

  public int range() {
    return cap * mpg;
  }
}

// client code
GasTank g1 = new GasTank(10, 20);
System.out.print(g1); // line 1

GasTank g2 = new GasTank(20, 15);
GasTank g3 = new GasTank(20, 15);
boolean b1 = g2 == g3;
boolean b2 = g3.equals(g2);
System.out.print(b1 + " " + b2); // line 2
```

QUESTION 19

Which of the following Java expressions is equivalent to the formula to the right? `a` and `t` are variables of type `double`.

A. `0.5 * a * t * t`

B. `a * t ^ 2 / 2`

C. `a * t * t * 2`

D. `(a * t * t) / 0.5`

E. `a * t ^^ 2 * 0.5`

$$\frac{1}{2}at^2$$

QUESTION 20

What is output by the code to the right?

A. `0` B. `3` C. `5`

D. `15` E. `null`

```
int[][] table = new int[5][3];
System.out.print(table[0].length);
```

QUESTION 21

What is output by the code to the right?

A. `1.0` B. `-1.0` C. `-2.0`

D. `2` E. `2.0`

```
double a3 = 17.02 * -0.1;
System.out.print(Math.floor(a3));
```

QUESTION 22

What is output by the code to the right?

A. `c84k c84k` B. `C*$K c84k`

C. `c*$K c*$K` D. `C84K c84k`

E. `c84k C84K`

```
String an1 = "C84K";
String an2 = an1.toLowerCase();
System.out.print(an1 + " " + an2);
```

QUESTION 23

What is output by the code to the right?

A. `[C, X, D]` B. `[X, X, D]`

C. `[C, X, X]` D. `[C, X, C]`

E. There is no output due to a syntax error in the code.

```
List<String> list;
list = new ArrayList<String>();
list.add("C");
list.add("D");
list.add(1, "X");
list.set(2, list.get(1));
System.out.print(list);
```

QUESTION 24

What is output by the client code to the right?

A. `48 6 24`

B. `48 12 24`

C. `48 6 12`

D. `24 6 24`

E. `48 6 48`

```
public int mystery(int x, int y) {
  x *= 2;
  y /= 3;
  return x * y;
}

// client code
int x = 6;
int y = 12;
System.out.print(mystery(x, y));
y = mystery(y, x);
System.out.print(" " + x + " " + y);
```

QUESTION 25

Which of the following can replace **`<*1>`** in the following line of code so that it compiles without error?

```
double <*1> = 0.0;
```

A. `37_` B. `Big` C. `final` D. `int` E. None of A through D are correct.

QUESTION 26

Given method `calc` to the right what is output by the following client code?

```
int[] data3 = {12, 14, -10, 4, 5, -10};
System.out.print(calc(data3));
```

A. `0` B. `2` C. `5`

D. `10` E. `26`

QUESTION 27

Given method `calc` to the right what is output by the following client code?

```
int[] data4 = {3, 1, 11, -3, 15, 1};
System.out.print(calc(data4));
```

A. `NaN` B. `0` C. `4`

D. There is no output due to a syntax error in the client code

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

```
public int calc(int[] list) {
  int total = 0;
  int count = 0;
  for(int i = 0; i < list.length; i++)
    if(list[i] % 2 == 0) {
      total += list[i];
      count++;
    }
  return total / count;
}
```

QUESTION 28

What is output by the code to the right?

A. `[A, B, aa, a]` B. `[aa, a, A, B]`

C. `[a, A, aa, B]` D. `[A, B, a, aa]`

E. `[A, a, aa, B]`

```
String[] sts = {"A", "a", "aa", "B"};
Arrays.sort(sts);
System.out.print(Arrays.toString(sts));
```

QUESTION 29

What is output by the code to the right?

A. `03025` B. `01234` C. `12345`

D. `10` E. `250`

```
int[] data = {0, 3, 0, 2, 5};
String all = "";
for(int xv : data)
  all += xv;
System.out.print(all);
```

QUESTION 30

What is output by the code to the right?

A. `{-4=aa, 0=A, 3=A,B, 5=null}`

B. `{-4=aa, 3=B, 5=null}`

C. `{-4=aa, 0=A, 3=B, 5=null}`

D. There is no output due to a syntax error.

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

```
TreeMap<Integer, String> samp;
samp = new TreeMap<Integer, String>();
samp.put(3, "A");
samp.put(0, "A");
samp.put(3, "B");
samp.put(-4, "aa");
samp.put(5, samp.get(-1));
System.out.print(samp);
```

QUESTION 31

A method is $O(N^3)$. When N = 200,000 the method takes 3 seconds to complete. What is the expected runtime for the method when N = 800,000?

A. 12 seconds B. 48 seconds C. 192 seconds D. 384 seconds E. 576 seconds

QUESTION 32

Given method `sort` to the right what is output when the following client code is executed.

```
int[] us = {12, 17, 5, -5, 15, 0, -3};
sort(us);
```

A. `[-5, 5, 12, 15, 17, 0, -3]`

B. `[-5, 0, -3, 5, 12, 15, 17]`

C. `[-5, -3, 0, 5, 12, 15, 17]`

D. `[12, 17, 5, -5, 15, 0, -3]`

E. None of A through D are correct.

```
public void sort(int[] lt) {
  int temp, j;
  for(int i = 1; i < lt.length; i++) {
    temp = lt[i];
    j = i;
    while( j > 0 && temp < lt[j - 1]) {
      lt[j] = lt[j - 1];
      lt[j - 1] = temp;
      j--;
    }
    if(i == 4)
     System.out.print(Arrays.toString(lt));
  }
}
```

QUESTION 33

Which sorting algorithm does method `sort` implement?

A. radix sort B. selection sort

C. insertion sort D. heap sort

E. None of A through D are correct.

QUESTION 34

Given method `gen` to the right what is output by the following client code?

```
ArrayList w = new ArrayList();
gen(w, 0, "eerily");
System.out.print(w.size());
```

A. `0` B. `14`

C. `64` D. `128`

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

```
public void gen(ArrayList pats,
                      int pos, String cur) {
  if(pos == cur.length())
    pats.add(cur);
  else {
    gen(pats, pos + 1, cur);
    String temp = cur.substring(0, pos) +
              'e' + cur.substring(pos + 1);
    gen(pats, pos + 1, temp);
  }
}
```

QUESTION 35

What is output by the code to the right?

A. `35.216` B. `16.804` C. `4.402`

D. `2.201` E. There is no output due to a syntax error.

```
double a5 = 8.804;
double b5 = a5 >> 2;
System.out.print(a5 + " " + b5);
```

QUESTION 36

Method `min` to the right contains a logic error. Which of the following changes is required so that method `min` always meets its post condition?

A. Change `int min = 0` to `int min = lt[0]`.

B. Change every occurrence of the identifier `min` to `m`.

C. Change `int min = 0` to `int min = Integer.MAX_VALUE`.

D. Change `int i` to `int x`.

E. More than one of A through D is correct.

```
// pre: lt != null, lt.length > 0
// post: return the minimum value in lt
public int min(int lt[]) {
  int min = 0;
  for(int i : lt)
    min = min > i ? i : min;
  return min;
}
```

QUESTION 37

What is returned by method `h(t, 1, 0, -10)` if `t` is the matrix shown below?

1	4	0	22	25	6
0	-1	5	20	47	50
2	3	7	17	10	52
11	5	13	15	4	20

A. 5 B. 8 C. 12

D. 14 E. 2000

```
public int h(int[][] t, int i, int j,
                                    int d) {
  if(i == -1 || i == t.length)
    return 1000;
  if(j == t[0].length)
    return 0;
  if(t[i][j] <= d)
    return 2000;
  int m1 = 1 + h(t, i - 1, j, t[i][j]);
  int m2 = 2 + h(t, i, j + 1, t[i][j]);
  int m3 = 1 + h(t, i + 1, j, t[i][j]);
  return Math.min(Math.min(m1, m2), m3);
}
```

QUESTION 38

Given class `N` below, what is output by the client code to the right?

```
public class N {
  public Object d;
  public N n;
  public N(Object d1, N n1) {
    d = d1; n = n1; }
}
```

A. 401 B. 302 C. 321

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

E. The output will vary from one run of the program to the next.

```
// client code
N n1 = new N(0, null);
n1.n = new N(1, new N(2, new N(3, n1)));
N n2 = new N(n1, new N(4, n1.n));
n2.n.n = n1.n.n.n;
N t = n2;
for(int i = 1; i < 11; i++) {
  if(i % 3 == 0)
    System.out.print(t.d);
  t = t.n;
}
```

QUESTION 39

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

```
Structure st = new Structure();
st.add(24);
st.add(36);
st.add(24);
st.add(8);
while(!st.isEmpty())
  System.out.print(st.remove());
```

A. 36824 B. 82436

C. 8243624 D. 2436248

E. The code runs without error, but there is no output.

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

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

  public void add(E val) {con.add(val);}

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

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

  public boolean isEmpty() {
    return con.isEmpty();
  }
}
```

QUESTION 40

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

A. a graph B. a heap C. a stack

D. a queue E. a list

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

interface java.util.Queue<E>
- boolean add(E e)
- boolean isEmpty()
- E peek()
- E remove()

class java.util.PriorityQueue<E>
- boolean add(E e)
- boolean isEmpty()
- E peek()
- E remove()

interface java.util.Set<E>
- boolean add(E e)
- boolean contains(Object obj)
- boolean remove(Object obj)
- int size()
- Iterator<E> iterator()
- boolean addAll(Collection<? extends E> c)
- boolean removeAll(Collection<?> c)
- 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>
- 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()

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

interface java.util.Iterator<E>
- boolean hasNext()
- E next()
- void remove()

interface java.util.ListIterator<E> extends java.util.Iterator<E>

Methods in addition to the `Iterator` methods:
- void add(E e)
- void set(E e)

class java.lang.Exception
- Exception()
- Exception(String message)

class java.util.Scanner
- Scanner(InputStream source)
- boolean hasNext()
- boolean hasNextInt()
- boolean hasNextDouble()
- String next()
- int nextInt()
- double nextDouble()
- String nextLine()
- Scanner useDelimiter(String pattern)

Computer Science Answer Key
UIL District 1 2012

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

Notes:

17. The GasTank inherits the toString method from the Object class, but the behavior of Object's toString is unpredictable.

27. An `ArithmeticException` occurs due to a divide by 0.

35. The `>>` operator may not be applied to `doubles`.

36. Choice A and C are both correct.

University Interscholastic League

Computer Science Competition

Number 134 (District 2 - 2012)

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 the sum of $D0A_{16}$ and 238_{16}?

A. $F42_{16}$ B. $AD2_{16}$ C. $E48_{16}$ D. $F48_{16}$ E. $E56_{16}$

QUESTION 2

What is output by the code to the right?

A. 4 B. 28 C. 56

D. 77 E. 84

```
int x = 4;
int y = x + 3 * (4 + x);
System.out.print(y);
```

QUESTION 3

What is output by the code to the right?

A. 0 B. 9 C. 10

D. 24 E. 27

```
int val = 0;
for(int i = 10; i > 1; i--)
  val += 3;
System.out.print(val);
```

QUESTION 4

What is output by the code to the right?

A. 13 B. 12 C. 7

D. 6 E. -1

```
String c1 = "AABABBAAABBBAAABB";
int res = c1.indexOf("AAA", 8);
System.out.print(res);
```

QUESTION 5

What is output by the code to the right?

A. 1 2 B. 3 2 C. 65 12

D. 2 1 E. 2 3

```
String[] st = {"AB", "A", "DAD", "12"};
System.out.print(st[3].length());
System.out.print(" " + st[1].length());
```

QUESTION 6

What is output by the code to the right?

A. 158 B. 96 C. 64

D. 32 E. 19

```
int x1 = 13;
int y1 = 19;
int z1 = 2 * x1 - y1 + 2 * y1 - x1;
System.out.print(z1);
```

QUESTION 7

How many combinations of values for the `boolean` variables `p`, `q`, and `r` will result in `s` being set to `true`?

A. 1 B. 2 C. 3

D. 4 E. 7

```
boolean p, q, r;
//code to initialize p, q, and r

boolean s = (p || q) && (!r && !q);
```

QUESTION 8

What is output by the code to the right?

A. `43.85` B. `13.85`

C. `47.7` D. `17.7`

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

```
double a3 = 7.7;
if(a3 / 2 > 3)
  System.out.print(1);
else
  System.out.print(4);
System.out.print(a3);
```

QUESTION 9

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

```
Student st1 = new Student(10);
st1.goodYear();
System.out.print(st1);
```

A. `credits: 6` B. `10`

C. `credits: 16` D. `6`

E. `credits: credits`

QUESTION 10

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

```
Student st2 = new Block(10, 4);
st2.goodYear();
System.out.print(st2);
```

A. `credits: 16` B. `credits: 14`

C. `credits: 10` D. `credits: 12`

E. There is no output due to a syntax error in the client code.

```
public class Student {
  private int credits;

  public Student(int c) {credits = c;}

  public void goodYear() {credits += 6;}

  public String toString() {
    return "credits: " + credits;
  }

  public void setCredits(int c) {
    credits = c;
  }
}

public class Block extends Student {
  private int ccs;

  public Block (int c, int ex) {
    super(c);
    ccs = ex;
  }

  public void goodYear() {
    setCredits(ccs + 8);
  }
}
```

QUESTION 11

What is output by the code to the right?

A. `189` B. `127` C. `62`

D. `31` E. `0`

```
int m = 127;
int n = 62;
System.out.print(m & n);
```

QUESTION 12

What is output by the code to the right?

A. `6.0` B. `5.0` C. `4.825`

D. `4.0` E. `3.15`

```
double m2 = 3.15;
double n2 = m2 / 2;
double o2 = Math.floor(n2) + Math.ceil(m2);
System.out.print(o2);
```

QUESTION 13

What is output by the code to the right?

A.
```
bigtallthin
```
B.
```
bigtall\nthin
```
C.
```
big
tall\nthin
```
D.
```
big
tallnthin
```
E.
```
big
tall
thin
```

```
System.out.print("big");
System.out.println();
System.out.print("tall\nthin");
```

QUESTION 14

What is output by the code to the right?

A. `671.5` B. `+671.4` C. `671.4`

D. `(671.0)` E. `+671.5`

```
double t5 = 671.45;
System.out.printf("%+6.1f", t5);
```

QUESTION 15

What is returned by the method call `calc(-2, -1.5)`?

A. `-31.5` B. `-21.0` C. `-15.0`

D. `-13.5` E. `3.5`

```
public double calc(int x, double a) {
  x--;
  a -= 2 * x;
  return a * x;
}
```

QUESTION 16

What is output by the code to the right?

A. `16` B. `15` C. `11`

D. `10` E. `0`

```
String stars = "";
for(int i = 0; i < 5; i++)
    stars += "*";
stars += "*";
for(int i = 0; i < 10; i++)
    stars += "*";
System.out.print(stars.length());
```

QUESTION 17

What is output by the code to the right?

A. `2` B. `2.0` C. `4.0`

D. `4` E. `5.0`

```
double[] as = {0.5, -1.78, 2.21, 4.5000};
System.out.print( (int) as[3] );
```

QUESTION 18

What is output by the code to the right?

A. `0` B. `25` C. `25.0`

D. `32` E. `32.0`

```
int r = 0;
r += Math.pow(2, 5);
System.out.print(r);
```

QUESTION 19

Which of the following Java statements is equivalent to the formula to the right? `BMI`, `ma`, and `in` are variables of type `double`.

A.
```
BMI = 703 * ma / (in * in);
```
B.
```
BMI = 703 * m * a / in / 2;
```
C.
```
BMI = 703.0 * ma / (in ^ 2);
```
D.
```
BMI = 703.0 * ma / in * 2;
```
E.
```
BMI = 703 * m * a >> in >> in;
```

$$BMI = \frac{703ma}{in^2}$$

QUESTION 20

What is the minimum value the code to the right will print out?

A. `0.0` B. `1.0` C. `2.0`

D. `3.0` E. None of A through D are correct.

```
double mys = 0.0;
double t = 0;
do {
  t = Math.random();
  mys++;
} while(t < 0.2);
System.out.print(mys);
```

QUESTION 21

What is output by the code to the right?

A. `true true` B. `false true`

C. `false false`

D. There is no output due to a syntax error.

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

```
Object obj1 = new ArrayList<String>();
Object obj2 = obj1.toString();
System.out.print(obj1 == obj2);
System.out.print(" " + obj1.equals(obj2));
```

QUESTION 22

Which of the following `boolean` expressions replaces **<*1>** in method `sort` so that the body of the `if` statement is executed if the `String` at index `m` is greater than the `String` at index `j`?

A. `w[m].compareTo(w[j]) > 0`

B. `w[m] > w[j]`

C. `w[m] < w[j]`

D. `w[m].compareTo(w[j]) < 0`

E. `Comparable.compareTo(w[m], w[j]) > 0`

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

QUESTION 23

What is output when the client code to the right is executed?

A. `[Z, a, MY, Z, b, p, BE]`

B. `[BE, MY, Z, Z, a, b, p]`

C. `[BE, MY, Z, Z, a, p, b]`

D. `[Z, Z, BE, MY, b, p, a]`

E. `[BE, MY, Z, Z, b, p, a]`

QUESTION 24

Which sorting algorithm does method `sort` implement?

A. insertion sort B. selection sort C. radix sort

D. quicksort E. heap sort

```
public void sort(String[] w) {
  for(int i = 0; i < w.length - 1; i++) {
    int m = i;
    for(int j = i + 1; j < w.length; j++)
      if( <*1> )
        m = j;
    if(m != i) {
      String t = w[i];
      w[i] = w[m];
      w[m] = t;
    }
    if(i == 4)
      System.out.print(Arrays.toString(w));
  }
}

// client code
String[] ds = {"Z", "a", "MY", "Z", "b",
                                    "p", "BE"};
sort(ds);
```

QUESTION 25

Which of the following is a valid Java identifier?

A. `+12` B. `5x` C. `(val)` D. `x_y12` E. More than one of A through D is correct.

QUESTION 26

What replaces **<*1>** in the method `search` to the right so that the output of the client code to the right is `[0, 2, 4, 8]`?

A. `m[i] = t` B. `m[i].equals(t)`

C. `m[i] == t` D. `m[i].compareTo(t)`

E. More than one of A through D is correct.

```
public ArrayList<Integer> search(
                    String[] m, String t) {
  ArrayList<Integer> r;
  r = new ArrayList<Integer>();
  for(int i = 0; i < m.length; i++)
    if(<*1>)
      r.add(i);
  return r;
}

// client code
String[] n = {"A", "N", "A", "J", "A",
                    "M", "AA", "AAA", "A"};
String sch = n[6].substring(1);
System.out.print(search(n, sch));
```

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

QUESTION 27

Which searching algorithm does method `search` implement?

A. binary B. heap

C. radix D. sequential

E. insertion

QUESTION 28

Which of the following can replace **<*1>** in the code to the right so that the code segment compiles without error?

I. `Collection<Integer>`
II. `List<Integer>`
III. `Object`

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

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

```
<*1> list = new LinkedList<Integer>();
System.out.print(list.toString());
System.out.print(list.size());
```

QUESTION 29

What replaces **<*1>** in method mystery to the right to obtain the remainder of the integer division of `x` by `i`?

A. `rem` B. `~` C. `!`

D. `/` E. `%`

```
public int mystery(int x) {
  int num = 0;
  for(int i = 1; i <= x; i++)
    if( x <*1> i == 0 )
      num++;
  return num;
}

// client code
System.out.print(mystery(13));
System.out.print(" " + mystery(72));
```

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

QUESTION 30

What is output by the client code to the right?

A. `2 12` B. `13 72` C. `1 11`

D. `1 7` E. `0 10`

QUESTION 31

What is output by the code to the right?

A. `10001100` B. `01110011` C. `00000000`

D. `11110011` E. `11111111`

```
byte b1 = 12;
byte b2 = (byte) ~b1;
String bits = Integer.toBinaryString(b2);
bits = bits.substring(bits.length() - 8);
System.out.print(bits);
```

QUESTION 32

What is the average case order (Big O) of method `find` shown to the right, given the following kinds of `Maps`? `m.size()` = N and `can.length` = M. Pick the most restrictive correct set of answers.

	TreeMap	HashMap
A.	O(NM)	O(NM)
B.	O(NlogM)	O(NM)
C.	$O(MN^2)$	O(N)
D.	$O(M^N)$	O(M)
E.	O(MlogN)	O(M)

```
public int find(Map<String, Integer> m,
                                String[] can) {
  int res = 0;
  for(String k : can)
    if(m.containsKey(k))
      res++;
  return res;
}
```

QUESTION 33

What is output by the code to the right?

A. `4 56` B. `11 5` C. `9 5`

D. `13` E. `6 56`

```
String gar = "56+=78-412**32^";
String[] rs = gar.split("\\D");
System.out.print(rs.length + " " + rs[0]);
```

QUESTION 34

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

11, 3, 7, 8, 3, 6, 9, 11, 12

A. 12 11 9 8 7 6 3 B. 3 6 7 8 9 11 12 C. 11 3 7 6 8 9 11 12

D. 6 9 8 7 3 12 11 E. 11 3 7 6 8 9 12

QUESTION 35

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

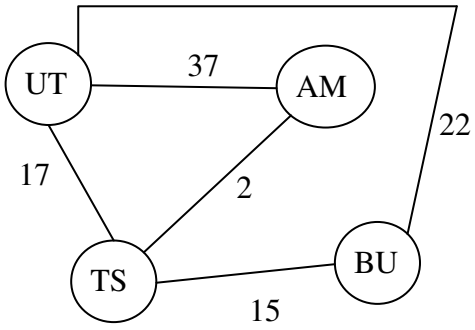
A. a directed unweighted graph

B. a directed weighted graph

C. an undirected unweighted graph

D. a undirected weighted graph

E. a binary search tree



QUESTION 36

What is output by the code to the right?

A. `truetruetrue`

B. `falsefalsetrue`

C. `falsefalsefalse`

D. `truefalsetrue`

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

```
String[] gds = {"A", "", null, "C", "D"};
System.out.print(gds instanceof Object);
System.out.print(gds[2] instanceof Object);
System.out.print(gds[1] instanceof String);
```

QUESTION 37

Given method `process` to the right, what is output by the following client code?

```
int[] p1 = process(9);
System.out.print(p1[4]);
```

A. 90 B. 9 C. 3

D. 2 E. 1

QUESTION 38

Given method `process` to the right, what is output by the following client code?

```
int[] p2 = process(15);
int tot = 0;
for(int i : p2)
  tot += i;
System.out.print(tot);
```

A. 105 B. 36 C. 33

D. 27 E. 14

```
public int[] process(int max) {
  int[] cs = {1, 3, 5};
  int[] ms = new int[max];
  Arrays.fill(ms, max * 10);
  ms[0] = 0;
  for(int i = 1; i < ms.length; i++)
    for(int j = 0; j < cs.length; j++)
      if(cs[j] <= i) {
        int t = ms[i - cs[j]] + 1;
        if(t < ms[i])
          ms[i] = t;
      }
  return ms;
}
```

QUESTION 39

What is output by the client code to the right?

A. 36912151821

B. 3612

C. 369121518

D. 181512963

E. 36120000000

QUESTION 40

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

A. An array based list

B. A stack

C. A set

D. A min heap

E. A queue

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

  public Structure() {
    con = new ArrayList<E>(10);
  }

  public void add(E v) { con.add(v); }

  public E remove() {
    return con.remove(0);
  }

  public E peek() { return con.get(0); }

  public boolean isEmpty() {
    return con.size() == 0;
  }
}

// client code
Structure<Integer> str;
str = new Structure<Integer>();
for(int i = 3; i < 20; i += i)
  str.add(i);
while(!str.isEmpty())
  System.out.print(str.remove());
```

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

interface java.util.Queue<E>
- boolean add(E e)
- boolean isEmpty()
- E peek()
- E remove()

class java.util.PriorityQueue<E>
- boolean add(E e)
- boolean isEmpty()
- E peek()
- E remove()

interface java.util.Set<E>
- boolean add(E e)
- boolean contains(Object obj)
- boolean remove(Object obj)
- int size()
- Iterator<E> iterator()
- boolean addAll(Collection<? extends E> c)
- boolean removeAll(Collection<?> c)
- 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>
- 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()

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

interface java.util.Iterator<E>
- boolean hasNext()
- E next()
- void remove()

interface java.util.ListIterator<E> extends java.util.Iterator<E>

Methods in addition to the `Iterator` methods:
- void add(E e)
- void set(E e)

class java.lang.Exception
- Exception()
- Exception(String message)

class java.util.Scanner
- Scanner(InputStream source)
- boolean hasNext()
- boolean hasNextInt()
- boolean hasNextDouble()
- String next()
- int nextInt()
- double nextDouble()
- String nextLine()
- Scanner useDelimiter(String pattern)

Computer Science Answer Key
UIL District 2 2012

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

Notes:

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.

31. Negative integers in Java are stored in 2's complement format.

33. When the delimiter is set without the plus sign (`"\\D"` instead of `"\\D+"`) sets of 2 delimiters in a row cause empty `String`s to be created.

University Interscholastic League

Computer Science Competition

Number 135 (Regional - 2012)

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 does $F9E_{16}$ minus 110011110001_2 equal?

A. $34D_{16}$ B. $1D8F_{16}$ C. 109_{10} D. $2AD_{16}$ E. $24F_{16}$

QUESTION 2

What is output by the code to the right?

A. 8 B. 108 C. 195

D. 895 E. 995

```
int x = 895;
int y = 100;
int z = x % y + y % x;
System.out.print(z);
```

QUESTION 3

What is output by the code to the right?

A. 16 B. 21 C. 23

D. 26 E. 28

```
int val = -5;
for(int i = val; i <= 8; i++){
  val++;
  ++val;
}
System.out.print(val);
```

QUESTION 4

What is output by the code to the right?

A. 20 B. 12 C. 9

D. 8 E. 6

```
String c1 = "#Yoo*";
String c2 = c1.toLowerCase();
c2 += c1 + c2 + c1;
System.out.print(c2.length());
```

QUESTION 5

What is output by the code to the right?

A. null5 B. 5 C. 4

D. There is no output due to a syntax error.

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

```
String[] st = new String[5];
System.out.print(st[3] + st.length);
```

QUESTION 6

What is output by the code to the right?

A. 2.0 B. 2.125 C. 5.125

D. There is no output due to a syntax error.

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

```
double a1 = 52.125;
a1 %= 10;
System.out.print(a1);
```

QUESTION 7

Which answer is logically equivalent to the following `boolean` expression, where `p`, `q`, and `r` are `int` variables?

```
!((p < q) || !(q >= r))
```

A. `(p < q) || (q >= r)` B. `(p >= q) && (q >=r)` C. `!(p < q) && !(q >= r)`

D. `p <= r` E. `!(p != q) && (q < r)`

QUESTION 8

What is output by the code to the right?

A. `12` B. `23`

C. `24` D. `14`

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

```
int x1 = 7;
if(x1 * 2 > 10)
  System.out.print(1);
else
  System.out.print(2);
if(x1 == 14)
  System.out.print(3);
else
  System.out.print(4);
```

QUESTION 9

What is output by statement marked `// line 1` in the client code to the right?

A. `3 1503`

B. `1500 4503`

C. `0 0`

D. `3 1500`

E. `3 1800`

QUESTION 10

What is output by statement marked `// line 2` in the client code to the right?

A. `2 300`

B. `2 400`

C. `2 600`

D. `2 800`

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

```
public class School {
  private int numStudents, cls;

  public School(int ns, int c) {
    numStudents = ns;
    cls = c;
  }

  public void newYear() {
    numStudents += cls * 100;
  }

  public String toString() {
    return cls + " " + numStudents;
  }
}

public class BigSchool extends School {
  public BigSchool(int ns, int c) {
    super(ns * 2, c);
  }

  public void newYear() {
    super.newYear();
    super.newYear();
  }
}

// client code
School sc1 = new School(1500, 3);
sc1.newYear();
System.out.print(sc1); // line 1
School sc2 = new BigSchool(100, 2);
sc2.newYear();
System.out.print(sc2); // line 2
```

QUESTION 11

What is output by the code to the right?

A. `54` B. `47` C. `43`

D. `35` E. `1`

```
int m = 35;
int n = 40;
int o = 19;
System.out.print(m & o | n);
```

QUESTION 12 Question 12 deleted due to error

Question 12 deleted due to error

e

QUESTION 13

What is output by the code to the right?

A. `"one`
`two`

B. `"one"two"`

C. `"onetwo`

D. `\t\onetwo`

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

```
System.out.print("\t\"one");
System.out.println("two");
```

QUESTION 14

What is output by the code to the right?

A. `100000000` B. `000256`

C. `000006` D. `000400`

E. `000000400`

```
int tk = 256;
System.out.printf("%06o", tk);
```

QUESTION 15

What is returned by the method call `change(7471020)`?

A. `0` B. `2` C. `21`

D. `747102` E. `247123`

```
public int change(int x) {
  if(x <= 0)
    return x;
  else
    return change(x / 10) + x % 10;
}
```

QUESTION 16

What is output by the code to the right?

A. `36` B. `37` C. `55`

D. `63` E. `64`

```
String stars = "*";
for(int i = 3; i < 12; i++)
  for(int j = 0; j < i; j++)
    stars += "*";
System.out.print(stars.length());
```

QUESTION 17

What is output by the code to the right?

A. `[0, 1, 2]` B. `[0.0, 1.0, 2.0]`

C. `[0, 0, 0]` D. `3 [0, 1, 2]`

E. The output will vary from one run of the program to the next.

```
int[] values = {0, 1, 2};
System.out.print(values);
```

QUESTION 18

How many combinations of values for the `boolean` variables `p`, `q`, `r`, and `s` will result in `t` being set to `true`?

A. `0` B. `1` C. `7`

D. `8` E. `15`

```
boolean p, q, r, s;

// code to initialize p, q, r, and s

boolean t = !p && q && r && !s;
```

QUESTION 19

What is output by statement marked `// line 1` in the client code to the right?

A. `0` B. `13` C. `15`

D. `18` E. `26`

QUESTION 20

What is output by statement marked `// line 2` in the client code to the right?

A. `0` B. `15` C. `18`

D. `26` E. `48`

```
public int tinker(int[] data) {
  data[0] += data[2];
  data[1] -= data[1] * 3;
  int t = 0;
  for(int i : data)
    t += i;
  data = new int[4];
  data[0] = 15;
  return t;
}

// client code
int[] readings = {12, 5, 7, 2};
int ans = tinker(readings);
System.out.print(ans); // line 1
ans = 0;
for(int i : readings)
  ans += i;
System.out.print(ans); // line 2
```

QUESTION 21

Which of the following can replace **`<*1>`** in the code to the right so that the code segment compiles without error?

I. `byte`
II. `short`
III. `float`

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

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

```
double value = 48.125627;
<*1> var = (<*1>) value;
```

QUESTION 22

Which of the following can replace **<*1>** in method `work` so that the method compiles without error?

A. `new` B. `work` C. `true`

D. `finally` E. `22_$`

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

```
public int work(int <*1>, int y) {
  int z = <*1> + y;
  <*1>++;
  y++;
  System.out.print(z + " ");
  return z + <*1> + y;
}

// client code
int x = 4;
int y = -5;
System.out.print(x++ + " "
               + work(x++, y) + " " + --y);
```

QUESTION 23

What is output by the client code to the right?

A. `1 5 3 -6` B. `0 4 2 -6`

C. `5 1 3 -5` D. `1 5 3 -4`

E. `4 0 2 -4`

QUESTION 24

What is output by the code to the right?

A. `2false`

B. `1false`

C. `1true`

D. The output will vary from one run of the program to the next.

E. There is no output due to a syntax error in the code.

```
boolean p = true;
if(p = false)
  System.out.print(1);
else
  System.out.print(2);
System.out.print(p);
```

QUESTION 25

What is the largest possible value the code to the right will output?

A. `990` B. `999` C. `1000`

D. `1089` E. `1100`

```
int tot = 0;
int lim = ((int) (Math.random() * 10)) + 1;
for(int i = 0; i < lim; i++) {
  int temp = (int) (Math.random() * 100);
  tot += temp;
}
System.out.print(tot);
```

QUESTION 26

Which of the following can replace **<*1>** in the code to the right so that the output is `6`?

A. `Character.isLowerCase(ch)`

B. `!Character.isLowerCase(ch)`

C. `Character.isLetter(ch)`

D. `!Character.isLetter(ch)`

E. `Character.isLetterOrDigit(ch)`

```
String uni = "Texas-Tech-2011";
int total = 0;
for(int i = 0; i < uni.length(); i++) {
  char ch = uni.charAt(i);
  if( <*1> )
    total++;
}
System.out.print(total);
```

QUESTION 27

What replaces **<*1>** in the code to the right so that when the `while` loop is complete `stck.size()` returns `0`?

A. `stck.pop()` B. `stck.isEmpty()`

C. `!stck` D. `!stck == 0`

E. `!stck.isEmpty()`

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

```
Stack<Integer> stck = new Stack<Integer>();
stck.push(-5);
stck.push(10);
if(stck.peek() > 0)
  stck.push(stck.peek());
while( <*1> )
  System.out.print(stck.pop());
```

QUESTION 28

What is output by the code to the right?

A. `1010-5` B. `-510`

C. `-51010` D. `10-5`

E. `10-5-5`

QUESTION 29

What is output by the statement to the right marked `// line 1`?

A. `false` B. `true` C. `-50`

D. There is no output due to a syntax error.

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

```
Comparable c1 = "Baylor";
Comparable c2 = "tcu";
boolean b3 = c1.compareTo(c2) > 0;

System.out.print(b3); // line 1

System.out.print(c1.equals(c2)); // line 2
```

QUESTION 30

What is output by the statement to the right marked `// line 2`?

A. `false` B. `true` C. `50`

D. There is no output due to a syntax error.

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

QUESTION 31

What is output by the code to the right?

A. `false false`

B. `false true`

C. `true false`

D. `true true`

E. The output will vary from one run of the program to the next.

```
TreeMap<Integer, String> tm;
tm = new TreeMap<Integer, String>();
tm.put(0, "A");
tm.put(12, "B");
tm.put(0, "C");
HashMap<Integer, String> hm;
hm = new HashMap<Integer, String>();
hm.put(0, "C");
hm.put(12, "" + 'B');
System.out.print(tm instanceof Collection);
System.out.print( " " + tm.equals(hm));
```

QUESTION 32

What is returned by the method call `test(7)`?

A. `-2` B. `19` C. `20`

D. `30` E. `38`

```
public int test(int x) {
  if(x <= 2)
    return x * 2;
  return x * 2 + test(x - 2) + test(x - 4);
}
```

QUESTION 33

What is the worst case order (Big O) of method `slide` to the right? N = `d1.length` and M = `d2.length`. Pick the most restrictive correct answer.

A. O(NM) B. O(NlogM)

C. O(N) D. O(MlogN)

E. $O(N^2)$

```
public int slide(int[] d1, int[] d2) {
  int res = 0;
  for(int i = 0; i < d1.length; i++)
    for(int j = i; j < i + 10; j++)
      if(j >= d2.length)
        break;
      else if(d2[j] > d1[i])
        res += d2[j];
  return res;
}
```

QUESTION 34

What is output by method `sort` when the following client code is executed?

```
int[] tst = {37, 52, 16, 8, 21, 53};
sort(tst);
```

A. `[21, 16, 8][53, 52, 37]`

B. `[16, 8][37, 21, 53, 52]`

C. `[37, 21, 53, 52, 16][8]`

D. `[8, 37][21, 53, 52, 16]`

E. `[53, 52, 37, 21, 16, 8][]`

QUESTION 35

What sorting algorithm does method `sort` implement?

A. selection sort

B. insertion sort

C. quicksort

D. merge sort

E. radix sort

```
// pre: all values in data > 0
public void sort(int[] data) {
  ArrayList<Integer>[] t =
   (ArrayList<Integer>[]) new ArrayList[2];
  t[0] = new ArrayList<Integer>();
  t[1] = new ArrayList<Integer>();
  int b = 1;
  for(int i = 0; i < 31; i++) {
    for(int j = 0; j < data.length; j++)
      t[(data[j] & b) / b].add(data[j]);
    b = b << 1;
    int j = 0;
    for(int x : t[1])
      data[j++] = x;
    for(int x : t[0])
      data[j++] = x;
    if(i == 4)
      System.out.print(t[0] + "" + t[1]);
    t[0].clear();
    t[1].clear();
  }
}
```

QUESTION 36

Assume method `regional(int[] data)` is $O(2^N)$ where N = `data.length`. When method `regional` is passed an array with `length` = 200 it takes 3 seconds for method `regional` to complete. If method `regional` is then passed an array with `length` = 207 what is the expected time it will take method `regional` to complete?

A. 3.1 seconds B. 128 seconds C. 384 seconds D. 1,024 seconds E. 3,072 seconds

QUESTION 37

Given method `prc` to the right what is output by the following client code?

```
int[] fs = {0, 8, 4, 12, 2, 10, 6, 14,
                          1, 9, 5, 13};

for(int i : prc(fs))
  System.out.print(i + " ");
```

A. `1 2 1 2 1 2 1 2 1 2 1 2`

B. `1 2 2 3 2 3 3 4 3 4 4 6`

C. `0 8 4 12 2 10 6 14 1 9 5 13`

D. `1 2 2 3 2 3 3 4 2 4 3 5`

E. `1 2 2 4 2 4 4 8 2 6 5 11`

```
public int[] prc(int[] data) {
  int[] f = new int[data.length];
  for(int i = 0; i < data.length; i++) {
    int m = 0;
    for(int j = 0; j < i; j++)
      if(data[i] > data[j] && f[j] > m)
        m = f[j];
    f[i] = m + 1;
  }
  return f;
}
```

QUESTION 38

Method `wrong` to the right has a syntax error. Which of the following best describes the syntax error?

A. `_` is not a valid identifier.

B. `C` is not a valid identifier for a variable.

C. The expression `C = 4` must be changed to `C == 4`.

D. The parameter `t` may not be declared to be `final`.

E. More than one of A through E is correct.

```
public void wrong(int[] _, final int t) {
  int C = 0;
  for(int wrong : _) {
    if(_[wrong] == t) {
      _[wrong]++;
      C++;
    }
    if(C = 4)
      return;
  }
}
```

GO ON TO THE NEXT PAGE.

QUESTION 39

What is output by the following client code?

```
Structure st = new Structure();
st.add(0, "A");
st.add(0, 12);
st.add(1, 0.5);
st.add(st.size(), "B");
for(int i = 0; i < st.size(); i++)
  System.out.print(st.get(i) + " ");
```

A. `12 0.5 B`

B. `A 0.5 12 B`

C. `12 0.5 A B`

D. There is no output due to a syntax error in the client code.

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

QUESTION 40

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

A. An array based list

B. A linked list

C. A stack

D. A queue

E. A graph

```
public class Structure<E> {
  private N<E> st = new N<E>(null, null);
  private int s;

  public void add(int i, E v) {
    N<E> n = new N<E>(v, g(i));
    g(i - 1).n = n;
    s++;
  }

  public E get(int i) { return g(i).d; }

  public void remove(int i) {
    g(i - 1).n = g(i).n;
    s--;
  }

  public int size() { return s; }

  private N<E> g(int i) {
    N<E> t = st;
    for(int j = -1; j < i; j++, t = t.n);
    return t;
  }

  private static class N<E> {
    private E d;
    private N<E> n;

    private N(E d1, N<E> n1) {
      d = d1;
      n = n1;
    }
  }
}
```

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

interface java.util.Queue<E>
- boolean add(E e)
- boolean isEmpty()
- E peek()
- E remove()

class java.util.PriorityQueue<E>
- boolean add(E e)
- boolean isEmpty()
- E peek()
- E remove()

interface java.util.Set<E>
- boolean add(E e)
- boolean contains(Object obj)
- boolean remove(Object obj)
- int size()
- Iterator<E> iterator()
- boolean addAll(Collection<? extends E> c)
- boolean removeAll(Collection<?> c)
- 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>
- 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()

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

interface java.util.Iterator<E>
- boolean hasNext()
- E next()
- void remove()

interface java.util.ListIterator<E> extends java.util.Iterator<E>

Methods in addition to the `Iterator` methods:
- void add(E e)
- void set(E e)

class java.lang.Exception
- Exception()
- Exception(String message)

class java.util.Scanner
- Scanner(InputStream source)
- boolean hasNext()
- boolean hasNextInt()
- boolean hasNextDouble()
- String next()
- int nextInt()
- double nextDouble()
- String nextLine()
- Scanner useDelimiter(String pattern)

Computer Science Answer Key
UIL Regional 2012

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

Notes: 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.

17. The hashcode of the variable is printed, which will vary from one run of the program to the next. Printing the contents of the array requires a call to `Arrays.toString(values)` or a loop to manually print each element.

31. `Map`s do not implement the `Collection` interface. `TreeMap`s and `HashMap`s are equal if they contain the same key-value pairs even though they may store them in different orders.

University Interscholastic League

Computer Science Competition

Number 136 (State - 2012)

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 does 10011_2 times 111_2 equal?

A. 11010_2 B. 1010_2 C. 1111100_2 D. 11110101_2 E. 10000101_2

QUESTION 2

What is output by the code to the right?

A. 0 B. 1

C. 2 D. 1.6666666666666667

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

```
double x = 3;
double y = 10;
y /= x / .5;
System.out.print((int)y);
```

QUESTION 3

What is output by the code to the right?

A. 500 B. 29

C. 20 D. 9

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

```
int j = 1;
int val = 0;
for(int i = 0; i<20 && j<500; i++, j*=2)
  val++;
System.out.print(val);
```

QUESTION 4

What is output by the code to the right?

A. 12base45 B. 4950base5152

C. abbasecd D. 3base9

E. 3base45

```
String c1 = "base";
String c2 = 1 + 2 + c1 + 4 + 5;
System.out.print(c2);
```

QUESTION 5

What is output by the code to the right?

A. [20, 1, 3, 4, 1, 2, 9]

B. [20, 1, 3, 4, 24, 2, 4]

C. [20, 1, 3, 4, 36, 2, 4]

D. There is no output due to a syntax error.

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

```
int[] st = {5, 1, 3, 4, 1, 2, 4};
st[st[st.length - 1]] += st[0];
st[4] *= st[st[4]];
st[0] *= st[3];
System.out.print(Arrays.toString(st));
```

QUESTION 6

What is output by the code to the right?

A. 288.0 B. 273.25 C. 272.0

D. 128.0 E. 83.0

```
double a1 = 0.125;
double b1 = 10 / a1 + (24 * a1);
System.out.print(b1);
```

QUESTION 7

How many combinations of values for the `boolean` variables `p`, `q`, `r`, and `s` will result in `t` being set to `false`?

A. 5 B. 7 C. 8

D. 9 E. 11

```
boolean p, q, r, s;
//code to initialize p, q, r, and s

boolean t = p || q && (!r || !s);
```

QUESTION 8

What is output by the code to the right?

A. 12 B. 13 C. 14

D. 23 E. 24

```
int x2 = 5;
int y2 = 6;
if(x2 > 0 || y2++ > 0)
  System.out.print(1);
else
  System.out.print(2);
if(y2 == 7)
  System.out.print(3);
else
  System.out.print(4);
```

QUESTION 9

Which of the following can replace **<*1>** in the code to the right so that method `hasMore` returns `true` if the value stored in the `credits` variable of the calling `Student` object is greater than the valued stored in the `credits` variable of `s2`, `false` otherwise?

I. `credits > s2.credits`

II. `this.getCredits() > s2.getCredits()`

III. `this.credits > s2.getCredits()`

A. II only B. I and II only

C. I and III only D. II and III only

E. I, II, and III

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

QUESTION 10

Which of the following can replace **<*2>** in the code to the right so that the client code compiles without error?

I. `hasMore(st2)`

II. `getCredits()`

III. `hasMore(st1)`

A. I only B. I and II only

C. I and III only D. II and III only

E. I, II, and III

```
public class Student {
  private int credits;

  public Student(int c) {credits = c;}

  public boolean hasMore(Student s2) {
    return <*1>;
  }

  public int getCredits() {
    return credits;
  }
}

// client code
Student st1 = new Student(12);
Student st2 = new Student(16);
boolean hm = st1.<*2>;
```

QUESTION 11

What is output by the code to the right?

A. 32 B. 37 C. 49

D. 52 E. 53

```
int m = 37;
int n = 57;
int o = 52;
System.out.print(o | m & n);
```

QUESTION 12

What is output by the code to the right?

A. 8.7 B. 9.0 C. 9.4

D. 9.7 E. 10.7

```
double m2 = -15.7;
double n2 = -6.3;
double o2 = Math.ceil(n2) + Math.abs(m2);
System.out.print(o2);
```

QUESTION 13

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

A. a4a-216-4 B. a4-a-216-5

C. a4a-216--4 D. a-2a416-4

E. a-2-a-24--2

```
public int ep(int a){
  System.out.print("a" + a);
  return a * a;
}

public void start(){
  System.out.print( ep(4) + "-" + ep(-2) );
}
```

QUESTION 14

What is output by the code to the right?

A. 37

B. .4

C. 0037.4

D. 37.424242

E. 37.420000

```
double d2 = 37.42;
System.out.printf("%2.6f", d2);
```

QUESTION 15

What is returned by the method call `notF(5)`?

A. 3 B. 11 C. 27

D. 46 E. 65

```
public int notF(int x) {
  return (x <= 0) ? 3 : notF(x-3) + x +
                                  notF(x-1);
}
```

QUESTION 16

What is output by the code to the right?

A. 2040 B. 2041 C. 2047

D. 2048 E. 4096

```
String stars = "";
for(int i = 8; i <= 1024; i *= 2)
  for(int j = 0; j < i; j++)
    stars += "*";
System.out.print(stars.length());
```

QUESTION 17

What is output by the code to the right?

A. `50`

B. `37`

C. `33`

D. `11`

E. `1`

```
String lets = "ABABC";
int res = 0;
for(int i = 0; i < lets.length(); i++) {
  char ch = lets.charAt(i);
  switch(ch) {
    case 'A': res += 2;
    case 'B': res += 3;
    case 'D': res += 4;
    default: res += 1; break;
  }
}
System.out.print(res);
```

QUESTION 18

What replaces **<*1>** in the code to the right so that the value stored in `val1` may not be altered once it is assigned?

A. `const` B. `static` C. `final`

D. `private` E. None of A - D are correct.

```
<*1> double val1;
double y1 = Math.random();
val1 = y1 * 100;
```

QUESTION 19

What is output by the code to the right?

A. `1234567890123`

B. `2147483647`

C. `1234567890123L`

D. There is no output due to a syntax error.

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

```
long bigVal = 1234567890123L;
System.out.print(bigVal);
```

QUESTION 20

What is returned by method `readSome` to the right, if `sc` is connected to a file with the following data?

```
2 6 3 2 ABABAB
1 1 12.323 .2 1
1 3 1 1 32 14 145
```

A. `13` B. `16` C. `21`

D. `213` E. `353`

```
public int readSome(Scanner sc) {
  int res = 0;
  for(int i = 0; i < 10; i++) {
    while(!sc.hasNextInt())
      sc.next();
    res += sc.nextInt();
  }
  return res;
}
```

Go on to the next page.

QUESTION 21

What is output by the client code to the right?

A. `0 1 2-3` B. `0 3 6-1`

C. `0 1 2-2` D. `2 2 2-1`

E. `0 3 60 1 22 2 2-3`

QUESTION 22

What are the best case and worst case orders (Big O) of method `search`? N = `dt.length`. Pick the most restrictive, correct set of answers.

	Best Case	Worst Case
A.	O(logN)	O(logN)
B.	O(1)	O(1)
C.	O(logN)	$O(2^N)$
D.	O(1)	O(N)
E.	O(1)	O(logN)

QUESTION 23

Which searching algorithm do methods `search` and `help` implement?

A. linear search B. interpolation search

C. sequential search D. binary search

E. map search

```
public int search(int[] dt, int tgt) {
  return help(dt, tgt, 0, dt.length, 1);
}

public int help(int[] dt, int tgt,
                  int st, int en, int c) {
  int m = st + ((en - st) / 2);

  // start debug section
  if(c == 2)
    System.out.print(st+" "+m+" "+en);
  // end debug section

  if(st > en)
    return -st - 1;
  else if(dt[m] < tgt)
    return help(dt, tgt, m + 1, en, c + 1);
  else if(dt[m] > tgt)
    return help(dt, tgt, st, m - 1, c + 1);
  else
    return m;
}

// client code
int[] data = {-6, -3, -1, 0, 5, 6, 9};
System.out.print(search(data, -2));
```

QUESTION 24

What is output by the code to the right?

A. `-5 21 -5 12`

B. `21 12 -5 -5`

C. `-5 12 21`

D. `21 12 -5`

E. `-5 -5 12 21`

```
PriorityQueue<Integer> pq;
pq = new PriorityQueue<Integer>();
pq.add(-5);
pq.add(21);
pq.add(-5);
pq.add(12);
while(!pq.isEmpty())
  System.out.print(pq.remove() + " ");
```

QUESTION 25

What is output by the code to the right?

A. `13` B. `23`

C. `19` D. `29`

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

```
int x3 = 3;
int y3;
if((y3 = x3) == 3)
  System.out.print(1);
else
  System.out.print(2);
if(x3 == y3)
  System.out.print(x3);
else
  System.out.print(x3 * y3);
```

QUESTION 26

Which of the following is not a Java keyword?

A. `float` B. `long` C. `String` D. `instanceof` E. `byte`

QUESTION 27

Given class `Grade` to the right, what is output by the line marked `// line 1` in the client code to the right?

A. `B+` B. `A+` C. `null`

D. There is no output due to a syntax error in the line in the client code marked `// line 1`.

E. The output will vary from one run of the program to the next.

QUESTION 28

Given class `Grade` to the right, what is output by the line marked `// line 2` in the client code to the right?

A. `C-` B. `Object` C. `null`

D. There is no output due to a syntax error in the line in the client code marked `// line 2`.

E. The output will vary from one run of the program to the next.

QUESTION 29

Given class `Grade` to the right, what is output by the line marked `// line 3` in the client code to the right?

A. `A+, A, A-`

B. `A, A+, A-`

C. `A, A-, A+`

D. There is no output due to a syntax error in the line in the client code marked `// line 3`.

E. There is no output due to a runtime error in client code section 3.

```
public class Grade {
  private String symbol;

  public Grade(String s) { symbol = s;}

  public String toString() {
    return symbol;
  }
}

// client code section 1
String str = "B+";
Grade g1 = new Grade(str);
str = "A+";
System.out.print(g1); // line 1


// client code section 2
Object ob2 = new Grade("C-");
System.out.print(ob2.toString()); // line 2


// client code section 3
TreeSet<Grade> ts;
ts = new TreeSet<Grade>();
ts.add(new Grade("A+"));
ts.add(new Grade("A"));
ts.add(new Grade("A-"));
System.out.print(ts.toString()); // line 3
```

QUESTION 30

What is output by the line marked `// line 1` in the client code to the right?

A. `17` B. `12` C. `2`

D. `-5` E. `-10`

QUESTION 31

What is output by the line marked `// line 2` in the client code to the right?

A. `0` B. `1` C. `3`

D. `31` E. `65`

```
public int rec2(int[] d, int s, int[] c) {
  c[0]++;
  if(s == 0)
    return d[0];
  else if(d[s] < rec2(d, s - 1, c))
    return d[s];
  else
    return rec2(d, s - 1, c);
}

// client code
int[] dat = {5, 2, -5, 3, 5, 12, -10, 17};
int[] c = new int[1];
int res2 = rec2(dat, 5, c);
System.out.print(res2); // line 1
System.out.print(c[0]); // line 2
```

QUESTION 32

Method `countLines` shown to the right will not compile due a syntax error. Which of the following changes will allow the method to compile without error?

I. Change the method header to

```
public int countLines(String f)
    throws FileNotFoundException {
```

II. Add this code after the line marked `// 3`

```
if(FileNotFound()) System.exit();
```

III. Add this code after the line marked `// 1`

```
try {
```

and add this code after the line marked `// 7`

```
} catch(FileNotFoundException fnf) {
  c = -1;
}
```

A. III only

B. I and II only

C. I and III only

D. II and III only

E. I, II, and III

```
public int countLines(String f) {
  int c = 0; // 1
  Scanner sc;
  sc = new Scanner(new File(f)); // 3
  while(sc.hasNextLine()) {
    c++;
    sc.nextLine();
  } // 7
  return c;
}
```

QUESTION 33

What is output by the code to the right?

A. `0 4`

B. `1 3`

C. `2 2`

D. `4 0`

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

```
int sm, dif;
sm = dif = 0;
for(int i = 0, g = 25; i < 2; i++, g += 30)
  for(int j = 0,h = 16; j < 2; j++, h*=2) {
    if(g % h == (g & (h - 1)))
      sm++;
    else
      dif++;
  }
System.out.print(sm + " " + dif);
```

QUESTION 34

What is output by the code to the right?

A. `[3, 2]` B. `[2, 3]` C. `[1, 3, 2]`

D. There is no output due to a syntax error.

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

```
HashMap<String, int[]> hm2;
hm2 = new HashMap<String, int[]>();
hm2.put("A", new int[]{3, 2});
hm2.get("A")[1]++;
hm2.get("A")[0]--;
String sth = Arrays.toString(hm2.get("A"));
System.out.print(sth);
```

Go on to the next page.

QUESTION 35

Based on the timing data for methods `sort1` and `sort2` to the right, which sorting algorithms do methods `sort1` and `sort2` implement? `sort1` and `sort2` both sort an array of `int`s into ascending order.

	`sort1` algorithm	`sort2` algorithm
A.	merge sort	quicksort
B.	merge sort	selection sort
C.	quicksort	selection sort
D.	quicksort	merge sort
E.	merge sort	insertion sort

Time to sort 1,000,000 distinct elements in random order:
```
sort1: 4 seconds
sort2: 3 seconds
```

Time to sort 4,000,000 distinct elements in random order:
```
sort1: 17.6 seconds
sort2: 13.2 seconds
```

Time to sort 250,000 distinct elements in descending order:
```
sort1: 0.9 seconds
sort2: 60 seconds
```

Time to sort 1,000,000 distinct elements in descending order:
```
sort1: 4 seconds
sort2: 960 seconds
```

QUESTION 36

What can replace **<*1>** and **<*2>** in the code to the right so that the entire code segment compiles without error?

	<*1>	**<*2>**
A.	`ArrayList`	`ListIterator`
B.	`List`	`Iterator`
C.	`LinkedList`	`Iterator`
D.	`Collection`	`ListIterator`

E. More than one of A through D is correct.

Assume **<*1>** and **<*2>** are filled in correctly.

```
String sd = "cs429hI";
List<Character> cList;
cList = new <*1><Character>();
for(int i = 0; i < sd.length(); i++)
  cList.add(0, sd.charAt(i));

<*2><Character> it = cList.listIterator();
int index = 0;
while(it.hasNext())
  if(Character.isLetter(it.next()))
    it.add(sd.charAt(index++));
  else
    it.set(sd.charAt(index));
for(char ch : cList)
  System.out.print(ch);
```

QUESTION 37

What is output by the code to the right?

A. `ccss429hhII`

B. `Ichs444s4c2`

C. `Ichs429shcI`

D. `ccss444h4I2`

E. `Ish4444s2c9`

Go on to the next page.

QUESTION 38

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

```
Structure str1 = new Structure();
int[] sData1 = {11,14,9,15,9,12,16,10};
for(int i : sData1)
  str1.add(i);
str1.show();
```

A. `0 16 15 14 12 11 10 9 9`

B. `16 14 15 11 9 9 12 10`

C. `16 15 14 12 11 9 9 10`

D. `16 15 14 12 11 10 9`

E. `16 15 14 12 11 10 9 9`

QUESTION 39

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

```
Structure str2 = new Structure();
int[] sData2 = {3,7,15,8,3,5,6,10};
for(int i : sData2)
  str2.add(i);
while(!str2.isEmpty())
  str2.remove();
System.out.print(str2.getCt());
```

A. `15 10 8 7 6 5 3 3 8`

B. `15 10 8 7 6 5 3 3 0`

C. `7`

D. `10`

E. `9`

QUESTION 40

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

A. a binary search tree

B. a min heap

C. a hash table

D. an array based list

E. a max heap

```
public class Structure {

  private int s;
  private int[] con;
  private int ct;

  public Structure() {
    con = new int[2];
  }

  public void add(int x) {
    if ( s >= con.length - 1 ) {
      int[] t = new int[con.length*2 + 1];
      System.arraycopy(con, 1, t, 1, s);
      con = t;
    }
    s++;
    int i = s;
    while ( i > 1 && x > con[i / 2]) {
      con[i] = con[i / 2];
      i /= 2;
    }
    con[i] = x;
  }

  public void show() {
    for(int i = 1; i <= s; i++)
      System.out.print(con[i] + " ");
  }

  public int remove() {
    int r = con[1];
    int x = 1;
    boolean d = false;
    while ( x * 2 < s && !d ) {
      ct++;
      int y = x * 2;
      if(con[y] < con[y + 1])
        y++;
      if(con[s] < con[y]) {
        con[x] = con[y];
        x = y;
      }
      else d = true;
    }
    con[x] = con[s];
    s--;
    return r;
  }

  public boolean isEmpty(){ return s == 0;}

  public int getCt() { return ct; }
}
```

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

interface java.util.Queue<E>
- boolean add(E e)
- boolean isEmpty()
- E peek()
- E remove()

class java.util.PriorityQueue<E>
- boolean add(E e)
- boolean isEmpty()
- E peek()
- E remove()

interface java.util.Set<E>
- boolean add(E e)
- boolean contains(Object obj)
- boolean remove(Object obj)
- int size()
- Iterator<E> iterator()
- boolean addAll(Collection<? extends E> c)
- boolean removeAll(Collection<?> c)
- 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>
- 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()

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

interface java.util.Iterator<E>
- boolean hasNext()
- E next()
- void remove()

interface java.util.ListIterator<E> extends java.util.Iterator<E>

Methods in addition to the `Iterator` methods:
- void add(E e)
- void set(E e)

class java.lang.Exception
- Exception()
- Exception(String message)

class java.util.Scanner
- Scanner(InputStream source)
- boolean hasNext()
- boolean hasNextInt()
- boolean hasNextDouble()
- String next()
- int nextInt()
- double nextDouble()
- String nextLine()
- Scanner useDelimiter(String pattern)

Computer Science Answer Key
UIL State 2012

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

Notes: 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.

8. Because the first part of the `boolean` expression `x2 > 0 || y2++ > 0` evaluates to `true` the expression will evaluate to `true` and the `||` operator short circuits. The second part of the expression, `y2++ > 0`, is not evaluated.

11. The `&` operator has a higher precedence than the `|` operator. Thus `m & n` is evaluated first.

17. Without `break` statements on the first three cases, fall through occurs until a `break` is found.

26. `String` is not a Java keyword. It may be used as an identifier. (The following code compiles: `int String = 12;`)

29. A runtime error (`ClassCastException`) occurs on the second call to `add` because the `Grade` class does not implement the `Comparable` interface.

36. The `Iterator` class does not have an `add` method. If the declared data type of `it` is `Iterator`, a syntax error occurs on the method call `it.add`.