QUESTION 1

What is the sum of binary numbers 101101_2 and 110001_2?

A. 1011110_2 B. 1101101_2 C. 1010110_2 D. 1110010_2 E. 1101010_2

QUESTION 2

What is output by the code to the right?

A. `No` B. `YesNo`

C. `Yes` D. `NoYes`

E. Nothing

```
int x = 10;
if (x < 100)
  System.out.print("Yes");
else
  System.out.print("No");
```

QUESTION 3

What replaces **`<*1>`** in the code to the right as the name of the static method that is executed when `MyClass` is executed by the Java interpreter?

A. `start` B. `init`

C. `main` D. `void`

E. `first`

```
public class MyClass {
  public static void <*1>(String[] args) {
    // code not shown
  }
}
```

QUESTION 4

What does `int[] intArray` look like after the static method call `process(intArray)` when `intArray` begins as the array below?

1	2	3	4	5	6

A.

1	2	3	4	5	6

B.

3	5	7	9	11	13

C.

3	6	9	12	15	18

D.

4	5	6	7	5	6

E. An exception is thrown

```
public static void process(int[] a) {
  for (int i=0; i < a.length; ++i)
    a[i] = a[i] * 2 + 1;
}
```

QUESTION 5

Which of these types is used to represent floating point numbers?

A. `int` B. `character` C. `double` D. `String` E. `ArrayList`

QUESTION 6

What replaces **`<*1>`** in the code to the right to indicate that `currentId` is a class variable shared by all instances of `Employee` and hidden from other classes?

A. `hidden static`

B. `public static`

C. `static`

D. `protected static`

E. `private static`

```
public class Employee {
  public Employee(String name) {
    this.name = name;
    this.id = currentId++;
  }
  private String name;
  private long id;
  <*1> int currentId = 0;
}
```

QUESTION 7

Assume **`<*1>`** is filled in correctly. What will be the `id` of the second `Employee` created by a program?

A. `0` B. `1`

C. `2` D. `3`

E. Cannot be determined

QUESTION 8

How many `*`'s are output by the code to the right?

A. `10` B. `0` C. `1`

D. `100` E. `99`

```
int x = 10, y = 100;
int z = y % x;
do System.out.print('*');
while (--z > 0);
```

QUESTION 9

Which of these expressions returns the value `"00"`?

A. `s.substring(2,3)`

B. `s.substring(1,3)`

C. `s.substring(1,2)`

D. `s.substring(2,2)`

E. `s.charAt(1) + s.charAt(1)`

```
String s = "2006 UIL Regional";
```

QUESTION 10

Suppose `x`, `y`, and `z` are initialized to `1`, `2`, and `3`, respectively. How many of the comparison operations are evaluated to get the value of `b`?

A. `2` B. `3` C. `4`

D. `5` E. `6`

```
int x, y, z;
// code to initialize x, y, and z not shown
boolean b =
  ((x < y) && (z < y)) ||
  ((y < z) || (z < x)) ||
  ((x >= z) && (y <= z));
```

QUESTION 11

What replaces `<*1>` in the code to the right to call the static method in the `Math` class that computes square roots to get the square root of `i`?

A. `Math m = new Math().sqrt(i)`

B. `Math.sqrt(i)`

C. `Math.root(i, 2)`

D. `Math.root(i, 1/2)`

E. `new Math().root(i, 0.5)`

QUESTION 12

Assume `<*1>` is filled in correctly. Assume `i` is positive and `r` is bigger than `2`. What is the running time of `root()`? Choose the most restrictive correct answer.

A. $O(i*r)$ B. $O(i^r)$

C. $O(r)$ D. $O(r^{1/i})$

E. $O(i^{1/r}*r)$

```
public static int findRoot(int i, int r) {
  if (r == 2)
    return (int) <*1>;
  else {
    for (int j=0; ;++j) {
      int n = 1;
      for (int k=0; k<r; ++k) {
        n *= j;
        }
      if (n > i) return j-1;
    }
  }
}
```

QUESTION 13

What replaces `<*1>` in the code to the right to give a random value between `1` and `n`, inclusive?

A. `r.nextInt(n) + 1`

B. `r.nextInt(n)`

C. `r.next(int) % n`

D. `r.nextInt() % n + 1`

E. Either A or D

QUESTION 14

Assume `<*1>` is filled in correctly. Which of these is the most likely to be output by the code below?

```
Dice d1 = new Dice(6),
     d2 = new Dice(10);

System.out.print(d1.roll() +
                 d2.roll());
```

A. `5` B. `8` C. `11`

D. B and C are equally likely

E. A, B, and C are equally likely

```
public class Dice {
  public Dice(int numSides) {
    n = numSides;
  }

  public int roll() {
    return <*1>;
  }

  private int n;
  private static Random r = new Random();
}
```

QUESTION 15

Which of these replaces **<*1>** in the code to the right to initialize data member `digits` to a `byte` array with length equal to `length`?

A. `digits.length() = length`

B. `digits.length = length`

C. `digits[length] = new byte[]`

D. `digits.length = new byte[length]`

E. `digits = new byte[length]`

QUESTION 16

Which of these replaces **<*2>** in the code to the right to set `i` to `0` if `neg` is `false` and `1` if `neg` is `true`?

A. `1 - neg` B. `(neg?1:0)`

C. `neg` D. `(neg?0:1)`

E. More than one of these

For the remaining questions, assume that **<*1>** and **<*2>** have been filled in correctly.

QUESTION 17

What value is `digits` initialized to if the first constructor is called with parameter `0`?

A. `0` B. `{0}`

C. `null` D. `{}`

E. A run-time error occurs

QUESTION 18

Which of these best describes the way `-38` would be stored in a `MyInteger` object?

A. In an array of length `2` with `-8` in the first position and `-3` in the second position

B. In an array of length `2` with `-3` in the first position and `8` in the second position

C. In an array of length `2` with `8` in the first position and `-3` in the second position

D. In an array of length `2` with `-3` in the first position and `-8` in the second position

E. As in answer A if created by the `int` constructor, but as in answer C if created by the `String` constructor

```
public class MyInteger {
  public MyInteger(int i) {
    if (i != 0) {
      int numDigits = 0, j = i;
      do {
        numDigits++;  j/=10;
      } while (j != 0);
      digits = new byte[numDigits];
      for (int k=0; k<numDigits; ++k) {
        digits[k] = (byte)(i%10);
        i/=10;
      }
    }
  }
  public MyInteger(String s) {
    if (!s.equals("0")) {
      boolean neg = false;
      if(s.charAt(0) == '-') neg = true;
      int length = s.length();
      if (neg) --length;
      <*1>;
      for (int i = <*2>; i < s.length();
                                       ++i) {
        char ch = s.charAt(i);
        if (ch < '0' || ch > '9')
          throw
            new IllegalArgumentException();
        digits[s.length()-i-1] =
               (byte)((ch-'0')*(neg?-1:1));
      }
    }
  }
  public String toString() {
    if (digits == null) return "0";
    else {
      String s = "";
      if (digits[0]>0)
        for (int i = digits.length-1; i>=0;
                                         --i)
          s += (char)(digits[i]+'0');
      else {
        s += '-';
        for (int i = digits.length-1; i>=0;
                                         --i)
          s += (char)(-digits[i]+'0');
      }
      return s;
    }
  }

  // arithmetic methods not shown

  private byte[] digits;

}
```

QUESTION 19

What is the output of the code to the right on the input below?

```
134 569 2abc
```

A. `1345692Not an integerErrorDone`

B. `134`

C. `134Done`

D. `2Error`

E. No output

QUESTION 20

What is the output of the code to the right on the input below?

```
cba2 965 431
```

A. `Not an integerDone`

B. `cbaNot an integerErrorDone`

C. `2`

D. `2Done`

E. `Not an integerErrorDone`

```
// nextInt() throws InputMismatchException
// when the next token is not an integer

Scanner in = new Scanner(System.in);
try {
  int x = in.nextInt();
  System.out.print(x);
}
catch(InputMismatchException e1) {
  System.out.print("Not an integer");
}
catch(Exception e2) {
  System.out.print("Error");
}
finally {
  System.out.print("Done");
}
```

QUESTION 21

What expression replaces **`<*1>`** in the code to the right to check whether the item at position `i` in `list` is the same as the `item` parameter?

A. `list[i] == item`

B. `list.get(i).equals(item)`

C. `list[i].equals(item)`

D. `((E)list.get(i)).equals<E>(item)`

E. `item.equals(list[i])`

QUESTION 22

Assume **`<*1>`** is filled in correctly. What is the running time of `search(li,it)` where `li` contains `n` items, none of which is `it`? Choose the most restrictive correct answer.

A. `O(1)`

B. `O(log n)`

C. `O(n)`

D. `O(n log n)`

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

```
public static <E> int
search(ArrayList<E> list, E item) {
  for (int i=0; i<list.size(); ++i)
    if (<*1>)
      return i;
  return -1;
}
```

QUESTION 23

What does `intArray` look like after the method call `process(intArray)` where `intArray` begins as the array below?

1	2	3
4	5	6
7	8	9

A.

1	2	3
4	5	6
7	8	9

B.

0	1	2
3	4	5
6	7	8

C.

0	1	2
1	2	3
2	3	4

D.

1	2	3
2	3	4
3	4	5

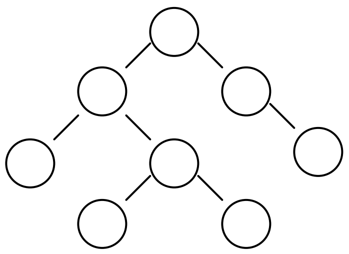
E.

0	0	0
0	0	0
0	0	0

```
public static void process(int[][] array) {
  for (int i=0; i<array.length; ++i)
    array[i] = new int[] {i, i+1, i+2};
}
```

QUESTION 24

Suppose the integers from `1` to `8` were inserted into a binary search tree. What order of insertion would generate the tree shape to the right?



A. `1 2 3 4 5 6 7 8`

B. `4 2 8 7 3 2 1 5`

C. `4 8 5 3 2 1 6 7`

D. `6 7 8 2 4 3 5 1`

E. `6 7 2 8 1 3 4 5`

QUESTION 25

What is the worst case running time of finding an item in a balanced binary tree with `n` elements? Choose the most restrictive correct answer.

A. `O(1)`
B. `O(log n)`
C. `O(n)`
D. `O(n log n)`
E. $O(n^2)$

QUESTION 26

What replaces **`<*1>`** in the code to the right to return `numStars` divided by `maxStars` and rounded to the nearest percent, that is, a number between `0` and `100`?

A. `100*Math.round(numStars/maxStars)`

B. `100*Math.floor(numStars/maxStars)`

C. `Math.floor(100*numStars/maxStars)`

D. `Math.round(100*numStars/maxStars)`

E. More than one of these

```
public class Rating {
  public Rating(String text,
           double numStars, int maxStars) {
    if (numStars>maxStars || numStars<0 ||
                                    maxStars<1)
      throw new IllegalArgumentException();
    this.text = text;
    this.numStars = numStars;
    this.maxStars = maxStars;
  }

  public int percentRating() {
    return (int)(<*1>);
  }

  private String text;
  private double numStars;
  private int maxStars;
}
```

For the remaining questions, assume that **`<*1>`** has been filled in correctly.

QUESTION 27

Suppose there is a user-defined class `Movie` which represents movies. Which of these is the declaration of a data structure that can be used to associate movies with lists of their ratings? Given a movie, the data structure must provide efficient lookup of its ratings.

A. `Map<List<Rating>,Movie> myRatings;`

B. `Map<Movie,List<Rating>> myRatings;`

C. `Movie<Map,Rating<List>> myRatings;`

D. `List<Map<Movie<Rating>>> myRatings;`

E. `List<Map<Movie,Rating>> myRatings;`

QUESTION 28

What is the output of the code below?

```
Rating r =
  new Rating("It was amazing!", 4.5, 5);
System.out.print(r.percentRating());
```

A. `It was amazing!` B. `4.5`

C. `90` D. `4`

E. `0`

QUESTION 29

What is returned by `process("12abCD")`?

A. `"12abCD"` B. `"12ABCD"`

C. `"CD"` D. `"  ABCD"`

E. The method does not compile

```
public static String process(String s) {
  StringBuffer sb = new StringBuffer();
  for (int i=0; i<s.length(); ++i)
    sb.append(Character.toUpperCase(
                              s.charAt(i)));
  return sb;
}
```

QUESTION 30

What sorting algorithm is implemented by the static method `Sorter.sort()`?

A. Selection sort B. Insertion sort

C. Quick sort D. Merge sort

E. The method does not correctly sort

QUESTION 31

Which of the following best describes what can be passed as a parameter to `Sorter.sort()`?

A. An array of items declared as `Object[]`

B. An array of items that implement the `IntKey` interface declared as `Object[]`

C. An array of items that implement the `IntKey` interface declared as `IntKey[]`

D. An array of items that implement the `IntKey` interface declared as `Type[]` where `Type` is any class that implements `IntKey`

E. Either C or D

QUESTION 32

Suppose that the array below is appropriate as a parameter to `Sorter.sort()`. Only the values returned by `key()` are shown rather than the full objects. What does the array look like after the third pass through the outer loop?

13	-4	10	5	6	18

A.

13	13	13	5	6	18

B.

-4	5	6	13	10	18

C.

-4	5	6	10	13	18

D.

-4	10	13	5	6	18

E.

13	-4	10	5	6	18

```
public interface IntKey {
  public abstract int key();
}

public class Sorter {
  public static int find(IntKey[] a,
            int front, int back, int key) {
    int i = front + (back - front)/2;
    while (front < back) {
      if (a[i].key()<key)
        front = i+1;
      else
        back = i;
      i = front + (back - front)/2;
    }
    return front;
  }

  public static void sort(IntKey[] a) {
    for (int i=1; i<a.length; ++i) {
      int pos = find(a,0,i,a[i].key());
      IntKey data = a[i];
      for (int j=i; j>pos; --j)
        a[j] = a[j-1];
      a[pos] = data;
    }
  }
}
```

QUESTION 33

What is returned by `Double.parseDouble("31e2")`?

A. `31.0` B. `3100.0` C. `310.0` D. `3.1` E. `0.31`

QUESTION 34

Which of these static method calls returns `4`?

A. `fish("1fish2fishredfishbluefish")`

B. `fish("1fish2fishredbluefishfish")`

C. `fish("1fish2fishredfishbluefishes")`

D. Both A and B

E. Both B and C

```
public static int fish(String s) {
  String[] array = s.split("fish");
  return array.length;
}
```

QUESTION 35

What replaces **`<*1>`** in the code to the right to call the constructor for class `A` with parameter `x`?

A. `super.x`   B. `this.x`

C. `super(x)`   D. `this(x)`

E. `A(x)`

QUESTION 36

What replaces **`<*2>`** in the code to the right to get the `String` returned by the `toString()` method from class `A`?

A. `A.toString`   B. `super.toString()`

C. `super`   D. `super.toString`

E. `"" + x`

```
public class A {
  public A(int x) {
    this.x = x;
  }
  public String toString() {
    return "" + x;
  }
  private int x;
}

public class B extends A {
  public B(int x, int y) {
    <*1>;
    this.y = y;
  }
  public String toString() {
    return <*2> + y;
  }
  private int y;
}
```

QUESTION 37

What replaces **`<*1>`** in the code to the right to add all of the numbers in `intArray` and store the result in `sum`?

A.
```
for (int i = 0; i<a.length; ++i)
   sum += i;
```

B. `for (int i : a) sum += i;`

C. `for (int i : a) sum += a[i];`

D. `sum += a;`

E. Either A or B

```
int[] a = {1, 2, 3, 4, 5};
int sum = 0;

<*1>
```

QUESTION 38

What is the output of the code to the right?

A. `%0(5.1f`   B. `-2.5678`

C. `(2.5678)`   D. `(02.6)`

E. An exception is thrown

```
double d = -2.5678;
System.out.printf("%0(6.1f", d);
```

QUESTION 39

What expression replaces **<*1>** in the code to the right, evaluating to `true` when `j` evenly divides `i`?

A. `(i % j != 0)`
B. `(j / i != 0)`
C. `(i % j == 0)`
D. `(j % i == 0)`
E. `(j / i == 0)`

QUESTION 40

Assume **<*1>** is filled in correctly. What is output by the code below?

```
FactorGame f = new FactorGame(50);
try {
  f.makeMove(47);
  f.makeMove(39);
  f.makeMove(45);
  f.makeMove(32);
  f.makeMove(10);
}
catch(Exception e) {}
System.out.print(f);
```

A.
```
Player 0: 230
Player 1: 100
The game is not over
```
B.
```
Player 0: 148
Player 1: 101
The game is over
```
C.
```
Player 0: 116
Player 1: 81
The game is not over
```
D.
```
Player 0: 0
Player 1: 100
The game is over
```
E.
```
Player 0: 98
Player 1: 162
The game is not over
```

```
public class FactorGame {

  public FactorGame(int max) {
    values = new int[max+1];
    for (int i=0; i<max+1; ++i)
      values[i] = -1;
  }

  public void makeMove(int i) {
    if (i <= 0 || i >= values.length ||
             values[i] != -1 || done)
      throw new IllegalArgumentException();
    values[i] = turn;
    score[turn]+=i;
    turn = 1-turn;
    done = true;
    for (int j = 1; j<=i/2; ++j)
      if (<*1> && values[j]==-1) {
        values[j] = turn;
        done = false;
        score[turn] += j;
      }
  }

  public String toString() {
    return "Player 0: " + score[0] + "\n"
         + "Player 1: " + score[1] + "\n"
         + "The game is "
         + (done?"":"not ") + "over\n";
  }

  private int[] values;
  private int turn;
  private int[] score = new int[2];
  private boolean done = false;

}
```

Computer Science Answer Key
UIL Regional 2006

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