

Question 1

What is the sum of $2A_{16}$, $1B_{16}$, and 101011_2 ?

- A. 168_8 B. 11010101_2 C. $6B_{16}$ D. 70_{16} E. 112_8

Question 8

What is the output of the code to the right?

- A. true true B. true false C. false true
D. false false E. true true false

```
boolean a = true;
boolean b = true;
System.out.print( !(a ^ b) + " ");
a = false;
System.out.print(a || b);
```

Question 21

What is the output of the code to the right?

- A. true false B. false false C. false true
D. true true E. No output due to a Syntax error.

```
ArrayList<Object> all;
all = new ArrayList<Object>();
ArrayList<String> al2;
al2 = new ArrayList<String>();

all.add(1);
all.add("v");
all.add('1');
all.add(al2);

System.out.print(all.equals(al2)+ " ");
System.out.print(all.get(3) == al2);
```

Question 28

The following values are added one at a time, in the order shown, to a binary search tree using the traditional insertion algorithm. What is the height of the resulting tree? The height of a binary tree is the length of its longest path.

15, 10, 11, -3, 17, 5, 19, 18, 16, 0, 6

- A. 3 B. 5 C. 4 D. 7 E. 6

Question 31

What is the output of the code to the right when given this 2-d array?

10	8	9	1	30	7	10
3	2	4	3	2	1	1
1	10	4	7	5	13	1
7	14	9	2	4	11	3

- A. 30 B. 19 C. 39
D. -2 E. 3

```
public int twodim(int[][] matr){
    boolean trigger = true;
    int result = 0;

    for(int q=matr.length-1; q>0; q--){

        for(int w=0; w<matr[q].length;
            w+=q){

            if(trigger == true){
                result+= matr[q][w];
            }else{
                result-=matr[q][w];
            }

            trigger = !trigger;
        }
    }
    return result;
}
```

1) D

$2A_{16}$ in base 10 is $2 * 16^1 + 10 = 42$

$1B_{16}$ in base 10 is $1 * 16^1 + 11 = 27$

101011_2 in base 10 is $2^5 + 2^3 + 2^1 + 2^0 = 32 + 8 + 2 + 1 = 43$

$42 + 27 + 43 = 112_{10}$

$16 * 7 = 112$ which will quickly lead you to the choice of D.

A is a garbage answer. Base 8 numbers contain only digits 0-7.

B is well over 112 by looking at the first digit.. $2^7 = 128$.

C might need to be worked out as it is close to the correct answer.

E is the correct digits in the wrong base.

8) A

a and b are both set to true in the beginning so $!(a \wedge b)$ will evaluate to true. The \wedge operator is Java's exclusive or meaning a or b.. but not both. So it's the same as $\|$ but $1 \wedge 1$ evaluates to false. a is then set to false and $a \| b$ is calculated. $0 \| 1$ evaluates to true.

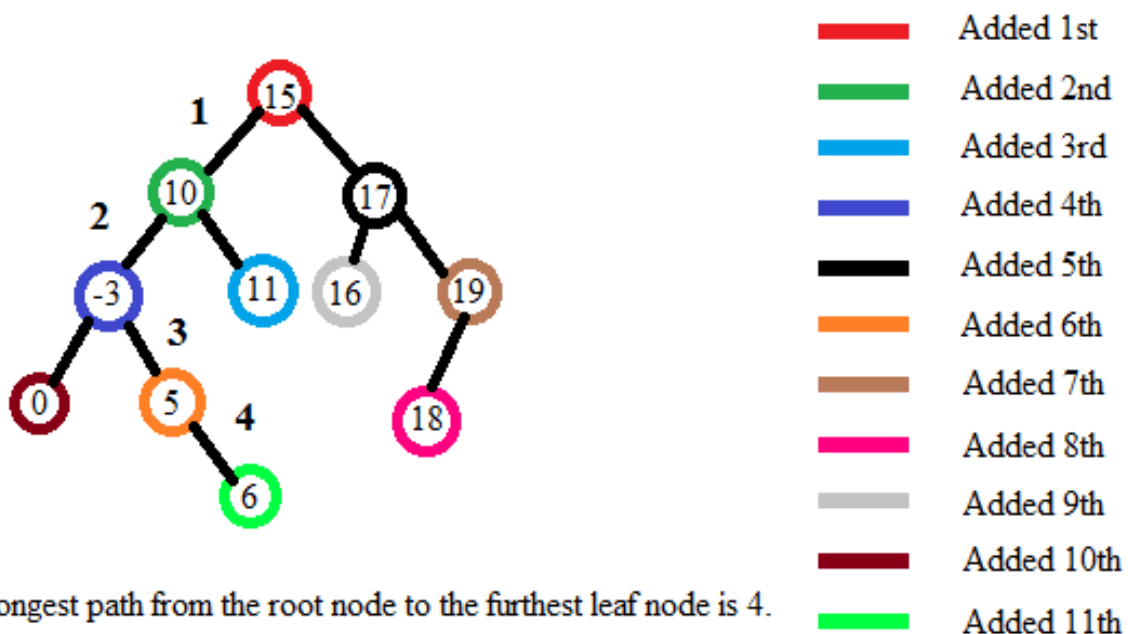
So, true true will be printed out by the code.

21) C

This code does not cause a Syntax error because Integer, Character, String, and ArrayList all inherit from the Object class, so they can all be put into an ArrayList containing objects.

It is obvious that $al1.equals(al2)$ will be false because al2 has nothing in it and contains objects of type String. When al2 is added into the ArrayList of objects, it is seen as an object inside of the list al1. Since, .equals compares the two arguments as objects this is fine and it will still return true when and Object with all the properties of an ArrayList is compared with itself as an ArrayList.

28) C



The longest path from the root node to the furthest leaf node is 4.

31) E

Think of a 2d array as an array of arrays. $\{\{data\}, \{data\}, \{data\}\}$

In this code, the first for loop is dealing with the length of the array containing the arrays.

The second for loop iterates through the actual arrays that are stored within the array.

The green items are added the red are subtracted.

10	8	9	1	30	7	10
3	2	4	3	2	1	1
1	10	4	7	5	13	1
7	14	9	2	4	11	3

The loop starts in the bottom left hand corner of the matrix, on the 7.

w is increased by q and whether you add or subtract switches every time.

q = 3 and w = 0

w = 0 res = 0 + 7: res = 7

w = 3 res = 7 - 2: res = 5

w = 6 res = 5 + 3: res = 8

q = 2 and w = 0

w = 0 res = 8 - 1: res = 7

w = 2 res = 7 + 4: res = 11

w = 4 res = 11 - 5: res = 6

w = 6 res = 6 + 1: res = 7

q = 1 and w = 0

w = 0 res = 7 - 3: res = 4

w = 1 res = 4 + 2: res = 6

w = 2 res = 6 - 4: res = 2

w = 3 res = 2 + 3: res = 5

w = 4 res = 5 - 2: res = 3

w = 5 res = 3 + 1: res = 4

w = 6 res = 4 - 1: res = 3