

Note: Correct responses are based on Java, J2sdk v 6.0, from Sun Microsystems, Inc. All provided code segments are intended to be syntactically correct, unless otherwise stated (i. e. `error` is an answer choice) and any necessary Java 2 Standard Packages have been imported. Ignore any typographical errors and assume any undefined variables are defined as used.

QUESTION 1	
What is 12_{11} plus 16_7 ?	
A. 26_{10} B. 11110_2 C. 34_7 D. 112_4 E. 27_9	
QUESTION 2	
What is output by the code to the right?	<pre>int a = 10; a = 5; a -= 3; out.println(a);</pre>
A. 12 B. -12 C. 8 D. -2 E. 2	
QUESTION 3	
What is output by the code to the right?	<pre>double c = 1.5; c += 7.1; out.println(c);</pre>
A. 7.6 B. 8.6 C. 5.1 D. 9.6 E. 7.1	
QUESTION 4	
What is output by the code to the right?	<pre>String s = "!!APLUS2001"; System.out.print(s.toLowerCase());</pre>
A. !!apulus2001 B. apulus C. !!APLUS2001 D. apulus2001 E. There is no output due to a runtime error.	
QUESTION 5	
What is output by the code to the right?	<pre>int e = 6 * 5 - 3 * 4; out.println(e);</pre>
A. 81 B. 18 C. 24 D. 28 E. 48	
QUESTION 6	
What is output by the code to the right?	<pre>String f = "thomasthetrain"; f=f.substring(0,f.indexOf('a')+1); out.println(f);</pre>
A. thomas B. thom C. asthe D. sthetrain E. thoma	
QUESTION 7	
What is output by the code to the right?	<pre>int c = 10; int g = 3; System.out.print(c % g + g % c);</pre>
A. 13 B. 5 C. 1 D. 3 E. 4	
QUESTION 8	
What is output by the code to the right?	<pre>int i = 'a'; i /= 2; out.println(i);</pre>
A. 2 B. 37 C. 48 D. 24 E. 65	
QUESTION 9	
What is output by the code to the right?	<pre>int j = 1236; int s = 0; while(j > 0){ s = s + j % 10; j = j / 10; } out.println(s);</pre>
A. 12 B. 11 C. 15 D. 10 E. 9	

<p>QUESTION 10</p> <p>What is output by the code to the right?</p> <p>A. 11 B. 4 C. 14 D. 0 E. 8</p>	<pre>int[][] k = new int[5][4]; k[1] = new int[6]; k[0][3] = 4; k[2][2] = 7; out.println(k[0][3] + k[4][2]);</pre>
<p>QUESTION 11</p> <p>What is the <i>last</i> value output by the code to the right ?</p> <p>A. 8.5 B. 11.0 C. 13.5 D. 16.0 E. 18.5</p>	<pre>for(double p = 1.0; p < 17; p += 2.5) out.println(p);</pre>
<p>QUESTION 12</p> <p>What is output by the code to the right?</p> <p>A. 0 B. 1 C. false D. true E. n</p>	<pre>boolean n = true; boolean o = true; out.println(! (n & o));</pre>
<p>QUESTION 13</p> <p>What is output by the code to the right?</p> <p>A. 0 B. 71 C. null D. 101 E. There is no output due to a runtime exception.</p>	<pre>String[] g = new String[101]; System.out.print(g[71].charAt(0));</pre>
<p>QUESTION 14</p> <p>Which of the following could correctly replace <*1> in the code to the right ?</p> <p>A. Boolean B. Long C. Stack D. A and B only E. A, B, and C</p>	<pre>ArrayList< <*1> > aList; aList = new ArrayList< <*1> >();</pre>
<p>QUESTION 15</p> <p>Which of the following could replace <*1> in the code to the right so method test will return true if parameter x is greater than 0 and less than 50?</p> <p>A. return x > 0 && x < 50; B. return !(x <= 0 x >= 50); C. return x > 0 x < 50; D. A and B only E. A, B, and C</p>	<pre>public boolean test(double x, int y) { <*1> }</pre>
<p>QUESTION 16</p> <p>What is output by the code to the right?</p> <p>A. 104 B. 82 C. 71 D. 77 E. There is no output due to runtime error</p>	<pre>String cat; cat = "NyanNYannyannyannyan"; cat = cat.replaceAll("[Nyan]" , "NYAN"); System.out.println(cat.length());</pre>
<p>QUESTION 17</p> <p>What is output by the code to the right?</p> <p>A. 7.591 B. 7.60 C. 7.590 D. 8.61 E. 7.61</p>	<pre>double u = 7.59123f; String v = String.format("%.3f",u); out.println(v);</pre>

QUESTION 18	
What is the output of // line 1?	
A. [3.6, 9.5, 1.8, 7.3]	
B. [3.6, 9.5, 1.8, 7.3, 5.6]	
C. [3.6, 9.5, 5.6, 1.8, 7.3]	
D. [5.6, 9.5, 1.8, 7.3, 3.6]	
E. [3.6, 5.6, 9.5, 1.8, 7.3]	
QUESTION 19	
What is the output of // line 2?	
A. 5.6	
B. 9.5	
C. 7.3	
D. 1.8	
E. There is no output due to a syntax error.	
QUESTION 20	
Which of the following is a sub-interface of Set?	
A. HashSet	B. TreeSet
C. SortedSet	D. Queue
E. more than one of these	
QUESTION 21	
What is output by the code to the right?	
A. 0	B. 1
C. true	D. false
E. There is no output due to syntax error.	
QUESTION 22	
What is output by the code to the right?	
A. 0	B. 1
C. -2	D. 2
E. There is no output due to a syntax error.	
QUESTION 23	
What correctly replaces <*1> in the code to the right so that all elements of chop are inspected?	
A. chop.nextInt()	B. chop.nextDouble()
C. chop.hasNextInt()	D. chop.hasNext()
E. chop.hasNextDouble()	
QUESTION 24	
Assuming <*1> was filled correctly, what is output by the code to the right?	
A. 20.1	B. 19.0
C. 18.1	D. 16.4
E. 17.1	

Stack<Double> who; who = new Stack<Double>(); who.add(9.5); who.push(1.8); who.add(0,5.6); who.push(7.3); who.add(0,3.6); out.println(who); who.pop(); who.pop(); out.println(who.peek());	// line 1 <
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<p>QUESTION 25</p> <p>What is the output of // line 1?</p> <p>A. 0 B. a C. z D. There is no output due to a syntax error. E. There is no output due to a runtime error.</p>	<pre>ArrayList<String> list; list = new ArrayList<String>(); for(int ii = 0; ii < 26; ii++) { list.add("" + (char)(ii+97)); } Iterator it = list.iterator(); System.out.println(it.next()); //line 1 it.remove(); while(it.hasNext()) { it.next(); it.remove(); } it = list.iterator(); it.next(); it.remove(); System.out.println(it.next()); //line 2</pre>
<p>QUESTION 26</p> <p>What is the output of // line 2?</p> <p>A. 0 B. a C. z D. There is no output due to a syntax error. E. There is no output due to a runtime error.</p>	<pre>Map<Integer, String> map; map = new TreeMap<Integer, String>(); map.put(7,"thomas"); map.put(7,"james"); map.put(6,"percy"); map.put(6,"gordon"); map.put(7,"emily"); map.put(7,"toby"); map.put(7,"diesel-10"); map.put(6,"henry"); map.put(6,"quazimobo"); out.println(map.get(6)); // line 1 out.println(map.get(7)); // line 2 out.println(map.get(8)); // line 3</pre>
<p>QUESTION 27</p> <p>What is the output of // line 1?</p> <p>A. henry B. quazimobo C. toby D. diesel-10 E. null</p>	<pre>public interface X { public static final int U = 76; } public class Man implements X { public static final int U = 23; } //////////////////////////////////// ///// client code X why = new Man(); System.out.print(why.U);</pre>
<p>QUESTION 28</p> <p>What is the output of // line 2?</p> <p>A. henry B. quazimobo C. toby D. diesel-10 E. null</p>	<p>QUESTION 29</p> <p>What is the output of // line 3?</p> <p>A. henry B. quazimobo C. toby D. diesel-10 E. null</p>
<p>QUESTION 29</p> <p>What is the output of // line 3?</p> <p>A. henry B. quazimobo C. toby D. diesel-10 E. null</p>	<p>QUESTION 30</p> <p>What is output by the code to the right?</p> <p>A. 0 B. 1 C. 76 D. 23 E. There is no output due to a syntax error.</p>
<p>QUESTION 30</p> <p>What is output by the code to the right?</p> <p>A. 0 B. 1 C. 76 D. 23 E. There is no output due to a syntax error.</p>	<pre>public interface X { public static final int U = 76; } public class Man implements X { public static final int U = 23; } //////////////////////////////////// ///// client code X why = new Man(); System.out.print(why.U);</pre>

QUESTION 31

Which of the following methods is called when attempting to insert a new Object into a HashSet?

- A. hashCode () B. toString () C. equalTo () D. compare ()
- E. more than one of these

QUESTION 32

Which of the following could replace **<*1>** in the code to the right so that method isIt () will terminate when i is equal to num?

- A. i <= num
B. i < num
C. i > num
D. A and B only
E. A, B, and C

```
public class Dino
{
    private int num;

    public Dino(int n)
    {
        num = n;
    }

    public boolean isIt()
    {
        int i = 1;
        int t = 0;
        while( <*1> )
        {
            if(num % i == 0)
                t = t + i;
            i++;
        }
        return ( t == num );
    }

    public String toString()
    {
        if(isIt())
            return "" + num + " is It.";
        else
            return "" + num + " is not It.";
    }
}
```

QUESTION 33

What is method isIt () trying to determine about num?

- A. is num a prime number
B. is num an odd number
C. is num a perfect number
D. is num an even number
E. is num a psuedo random number

QUESTION 34

What is the runtime efficiency of toString () ?
Choose the most restrictive correct answer.

- A. O(log₂N)
B. O(N²)
C. O(N)
D. O(N)*O(log₂N)
E. O(N²)*O(log₂N)

QUESTION 35

Assuming that **<*1>** was filled correctly, what is the output of the code at right?

- A. 496 is not It.
B. 496 is It.
C. ref
D. ref 496
E. There is no output due to a syntax error.

```
////////////////////////////////////
//client code
Dino ref;
ref = new Dino(496);
out.println(ref);           //line 1
```

QUESTION 36

What are the values of a after the code to the right has executed?

- A. [1, 5, 2, 4, -2, 7]
- B. [7, 5, 4, 2, 1, -2]
- C. [-2, 1, 2, 4, 5, 7]
- D. [7, 4, -2, 2, 1, 5]
- E. [5, 4, 2, 1, -2, 7]

```
public static int[] method()
{
    int[] a = new int[] {1,5,2,4,-2,7};
    for(int i=0;i<a.length;i++)
    {
        int min =i;
        for(int j=i+1;j<a.length;j++)
        {
            if(a[j]>a[min])    //line 1
                min = j;
        }
        a[min]=a[i]^a[min]^(a[i]=a[min]);
    }
    return a;
}
```

QUESTION 37

What values does a contain after the code to the right has executed the line marked //line 1 8 times?

- A. [7, 5, 2, 4, -2, 1]
- B. [1, 5, 2, 4, -2, 7]
- C. [7, 5, 4, 2, -2, 1]
- D. [7, 5, 2, -2, 1, 4]
- E. The code does not run this many times.

QUESTION 38

What string s would cause the code to the right to return true?

- A. True
- B. true
- C. False
- D. false
- E. NOPE!

```
Double Chuck = null;
Double Testa = null;
try{
    out.println(Chuck.equals(Testa));
} catch(Exception ee){
    out.println("NOPE!");
}
```

```

////////////////////////
///Client Code

Structure struct = new Structure(3);
struct.insert('a',0);
struct.insert('b',1);
struct.insert('c',2);
struct.insEd(0,2,100);
struct.insEd(1,2,76);
struct.insEd(0,0,29);
struct.insEd(0,1,58);
System.out.println(struct.toString());

```

QUESTION 39

What does the client code above output?

- | | |
|----------|----------|
| A. a 263 | B. a 188 |
| b 263 | b 268 |
| c 263 | c 176 |
| C. a 374 | D. a 188 |
| b 268 | b 268 |
| c 352 | c 352 |
| E. a 263 | |
| b 352 | |
| c 176 | |

QUESTION 40

What type of data structure is being created at right?

- A. priority queue
- B. weighted directed graph
- C. unweighted directed graph
- D. weighted undirected graph
- E. unweighted undirected graph

```

public class Structure
{
    int [][] brdgs;
    char[] vals;

    public Structure(int size){
        brdgs = new int[size][size];
        vals = new char[size];
    }

    public void insert(char c, int i){
        vals[i] = c;
    }

    public void insEd(int a,int b,int c){
        if(c>0) {
            brdgs[a][b] = c;
            brdgs[b][a] = c;
        }
    }

    public boolean remove(char c){
        int i,j;
        for(i=0;i<vals.length;i++){
            if(vals[i]==c){
                for(j=0;j<brdgs.length;j++){
                    brdgs[i][j] = 0;
                    brdgs[j][i] = 0;
                }
                vals[i] = 0;
                return true;
            }
        }
        return false;
    }

    public String toString(){
        String out = "";
        int i,j;
        for(i=0; i<vals.length; i++){
            int sum = 0;
            for(j=0; j<brdgs.length; j++){
                sum += brdgs[i][j]+brdgs[j][i];
                out += vals[i]+ " "+sum+"\n";
            }
            return out;
        }
    }
}

```