

UIL COMPUTER SCIENCE WRITTEN TEST

2024 DISTRICT

APRIL 2024

General Directions (Please read carefully!)

1. DO NOT OPEN THE EXAM UNTIL TOLD TO DO SO.
2. There are 40 questions on this contest exam. You will have 45 minutes to complete this contest.
3. All answers must be legibly written on the answer sheet provided. Indicate your answers in the appropriate blanks provided on the answer sheet. Clean erasures are necessary for accurate grading.
4. You may write on the test packet or any additional scratch paper provided by the contest director, but NOT on the answer sheet, which is reserved for answers only.
5. All questions have ONE and only ONE correct answer. There is a 2-point penalty for all incorrect answers.
6. Tests 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 test until told to do otherwise. You may use this time to check your answers.
7. If you are in the process of actually writing an answer when the signal to stop is given, you may finish writing that answer.
8. All provided code segments are intended to be syntactically correct, unless otherwise stated. You may also assume that any undefined variables are defined as used.
9. A reference to many commonly used Java classes is provided with the test, and you may use this reference sheet during the contest. AFTER THE CONTEST BEGINS, you may detach the reference sheet from the test booklet if you wish.
10. Assume that any necessary import statements for standard Java SE packages and classes (e.g., `java.util`, `System`, etc.) are included in any programs or code segments that refer to methods from these classes and packages.
11. NO CALCULATORS of any kind may be used during this contest.

Scoring

1. Correct answers will receive **6 points**.
2. Incorrect answers will lose **2 points**.
3. Unanswered questions will neither receive nor lose any points.
4. In the event of a tie, the student with the highest percentage of attempted questions correct shall win the tie.

STANDARD CLASSES AND INTERFACES – SUPPLEMENTAL REFERENCE

package java.lang

```
class Object
    boolean equals(Object anotherObject)
    String toString()
    int hashCode()

interface Comparable<T>
    int compareTo(T anotherObject)
        Returns a value < 0 if this is less than anotherObject.
        Returns a value = 0 if this is equal to anotherObject.
        Returns a value > 0 if this is greater than anotherObject.

class Integer implements Comparable<Integer>
    Integer(int value)
    int intValue()
    boolean equals(Object anotherObject)
    String toString()
    String toString(int i, int radix)
    int compareTo(Integer anotherInteger)
    static int parseInt(String s)

class Double implements Comparable<Double>
    Double(double value)
    double doubleValue()
    boolean equals(Object anotherObject)
    String toString()
    int compareTo(Double anotherDouble)
    static double parseDouble(String s)

class String implements Comparable<String>
    int compareTo(String anotherString)
    boolean equals(Object anotherObject)
    int length()
    String substring(int begin)
        Returns substring(begin, length()).
    String substring(int begin, int end)
        Returns the substring from index begin through index (end - 1).
    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 fromIndex. Returns -1 if str is not found.
    int indexOf(int ch)
    int indexOf(int ch, int fromIndex)
    char charAt(int index)
    String toLowerCase()
    String toUpperCase()
    String[] split(String regex)
    boolean matches(String regex)
    String replaceAll(String regex, String str)

class 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 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, int b)
    static int max(int a, int b)
    static long round(double a)
    static double random()
        Returns a double greater than or equal to 0.0 and less than 1.0.
```

package java.util

```
interface List<E>
class ArrayList<E> implements List<E>
    boolean add(E item)
    int size()
    Iterator<E> iterator()
    ListIterator<E> listIterator()
    E get(int index)
    E set(int index, E item)
    void add(int index, E item)
    E remove(int index)

class LinkedList<E> implements List<E>, Queue<E>
    void addFirst(E item)
    void addLast(E item)
    E getFirst()
    E getLast()
    E removeFirst()
    E removeLast()

class Stack<E>
    boolean isEmpty()
    E peek()
    E pop()
    E push(E item)

interface Queue<E>
class PriorityQueue<E>
    boolean add(E item)
    boolean isEmpty()
    E peek()
    E remove()

interface Set<E>
class HashSet<E> implements Set<E>
class TreeSet<E> implements Set<E>
    boolean add(E item)
    boolean contains(Object item)
    boolean remove(Object item)
    int size()
    Iterator<E> iterator()
    boolean addAll(Collection<? extends E> c)
    boolean removeAll(Collection<?> c)
    boolean retainAll(Collection<?> c)

interface Map<K,V>
class HashMap<K,V> implements Map<K,V>
class TreeMap<K,V> implements 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()

interface Iterator<E>
    boolean hasNext()
    E next()
    void remove()

interface ListIterator<E> extends Iterator<E>
    void add(E item)
    void set(E item)

class Scanner
    Scanner(InputStream source)
    Scanner(String str)
    boolean hasNext()
    boolean hasNextInt()
    boolean hasNextDouble()
    String next()
    int nextInt()
    double nextDouble()
    String nextLine()
    Scanner useDelimiter(String regex)
```

STANDARD CLASSES AND INTERFACES – SUPPLEMENTAL REFERENCE

Package `java.util.function`

Interface BiConsumer<T,U>
`void accept(T t, U u)`

Interface BiFunction<T,U,R>
`R apply(T t, U u)`

Interface BiPredicate<T,U>
`boolean test(T t, U u)`

Interface Consumer<T>
`void accept(T t)`

Interface Function<T,R>
`R apply(T t)`

Interface Predicate<T>
`boolean test(T t)`

Interface Supplier<T>
`T get()`

UIL COMPUTER SCIENCE WRITTEN TEST – 2024 DISTRICT

Note: Correct responses are based on **Java SE Development Kit 20 (JDK 20)** from Oracle, Inc. All provided code segments are intended to be syntactically correct, unless otherwise stated (e.g., "error" is an answer choice) and any necessary Java SE 20 Standard Packages have been imported. Ignore any typographical errors and assume any undefined variables are defined as used. **For all output statements, assume that the System class has been statically imported using: `import static java.lang.System.*;`**

Question 1

Find the product of 110011_2 and 64_{10}

- A) 6415_8 B) 6410_8 C) 775_8 D) 6300_8 E) 6214_8

Question 2

What is the output of the code segment to the right?

- A) -12 B) -8 C) -4 D) 1 E) 2

```
out.print(12 - 27 + 17 % 10 / 2);
```

Question 3

What is the output of the code segment to the right?

- A) 011
11
24
- B) 1411
10
24
- C) 1471
11
1
- D) 1471
10
24
- E) 1671
10
12

```
int A = 14;  
int B = 16;  
int C = 9;  
int D = 9;  
  
out.print(A % B);  
out.print(B % C);  
out.println(A / D);  
out.print(C / D);  
out.println(D % C);  
out.println(A + B / C + D);
```

Question 4

What is the output of the code segment to the right?

- A) A B) S C) C D) I E) T

```
String St1 = "KANSAS";  
String St2 = "CITY";  
String St3 = St2+St1.substring(0,3);  
out.print(St3.charAt(5));
```

Question 5

What is the output of the code segment to the right?

- A) true
B) false

```
boolean A = true;  
boolean B = !A;  
boolean C = A || B;  
boolean D = B && C;  
boolean E = A ^ D;  
out.print(E);
```

Question 6

What is the output of the code segment to the right?

- A) 18 B) 21 C) 24 D) 28 E) 35

```
int T = 0;  
for (int x=5; x<=10; x=x+2)  
    T+=x;  
out.print(T);
```

<p>Question 7</p> <p>What is the output of the code segment to the right?</p> <p>A) 2 B) 4 C) 8 D) 10 E) 16</p>	<pre>double A = Math.floor(8.7); double B = Math.ceil(A - 0.1); double C = Math.sqrt(A + B); int D = (int) C; out.print(D);</pre>
<p>Question 8</p> <p>What is the output of the code segment to the right?</p> <p>A) 14 B) 15 C) 16 D) 17 E) 18</p>	<pre>int A = 5; int B = A - 1; int C = B + 2; if (A > B) C++; if (C < A) B++; if (B < C) A++; out.print(A + B + C);</pre>
<p>Question 9</p> <p>What is the output of the code segment to the right?</p> <p>A) 6 B) 7 C) 8 D) 9 E) 10</p>	<pre>int[] Stuff = {4,2,1,0,3}; int A = Stuff[Stuff[2]]; int B = Stuff[Stuff[3]]; int C = Stuff[Stuff[0]]; out.print(A + B + C);</pre>
<p>Question 10</p> <p>What is the output of the code segment to the right?</p> <p>A) 60 B) 65 C) 70 D) 75 E) 80</p>	<pre>int[] elf = new int[50]; for(int x=0; x<=49; x++) elf[x] = x * 2 + 5; out.print(elf[10] + elf[20]);</pre>
<p>Question 11</p> <p>What is output by the code segment to the right?</p> <p>A) [7, 6, 2, 0, 2] B) [7, 5, 4, 3, 2] C) [7, 4, 5, 3, 2] D) [7, 2, 4, 0, 2] E) [1, 4, 0, 2, 2]</p>	<pre>ArrayList<Integer>Fred; Fred = new ArrayList<Integer>(); Fred.add(7); Fred.add(5); Fred.add(8); Fred.add(3); Fred.add(9); Fred.add(1); Fred.add(4); Fred.add(0); Fred.add(2); Fred.set(5, 2); Fred.add(4, 6); for (int x=1; x<=5; x++) Fred.remove(1); out.print(Fred);</pre>
<p>Question 12</p> <p>What is the output of the code segment to the right?</p> <p>A) -5 -4 -3 -2 -1 0 1 2 3 4 5 B) -5 -2 1 4 C) -10 -6 4 20 D) -5 -4 3 16 E) -5 -6 4 20</p>	<pre>for(int i=-5, T=1; i<=5; i=i+3, T++) out.print(i*T + " ");</pre>

<p>Question 13</p> <p>What is the output of the code segment to the right?</p> <p>A) 10 B) 12 C) 14 D) 15 E) 16</p>	<pre>int M = 6 + 10 & 4 + 10 << 2; out.print(M);</pre>
<p>Question 14</p> <p>What is the output of the code segment shown on the right?</p> <p>A) 22 B) 24 C) 42 D) 44 E) 84</p>	<pre>out.print(Long.SIZE/Short.SIZE); out.print(Integer.SIZE/Byte.SIZE);</pre>
<p>Question 15</p> <p>What is output by the code segment shown on the right?</p> <p>A) 5 B) 8 C) 10 D) 16 E) 20</p>	<pre>String Heat; Heat = "5 1 2 9 2 6 7 4 1 7"; Scanner Sc = new Scanner(Heat); int T = Sc.nextInt(); for (int x=0; x<=T; x++) Sc.next(); T += Sc.nextInt() + Sc.nextInt(); out.print(T);</pre>
<p>Question 16</p> <p>What is the output of the code segment shown on the right?</p> <p>A) 60 B) 66 C) 70 D) 77 E) 84</p>	<pre>int G = 0; for(int A=-3; A<=3; A++) for(int B=0; B<=100; B+=10) G++; out.print(G);</pre>
<p>Question 17</p> <p>What is the output of the code segment shown on the right?</p> <p>A) 25 B) 30 C) 35 D) 40 E) 45</p>	<pre>String Plasma = "QWERTYUIOP"; int R = 0; for(int x=0; x<Plasma.length()-1; x++) { char One =Plasma.charAt(x); char Two =Plasma.charAt(x+1); if(One<Two) R = R + 5; } out.print(R);</pre>
<p>Question 18</p> <p>What is the output of the code segment shown on the right?</p> <p>A) 42 B) 47 C) 48 D) 49 E) 53</p>	<pre>int A = 20; int B = (A>16) ? 21:15; int C = (A>B) ? 7:12; out.print(A + B + C);</pre>

Question 19

What is the output of the code segment shown on the right?

- A) 109
- B) 124
- C) 129
- D) 144
- E) 154

```
int[]Speed = {4,1,4,1,4,1};
int[]Velocity = {5,10,15,20,25,30,35,40};

int Texas = Speed[0];
for(int power: Speed)
    Texas += Velocity[power];
out.print(Texas);
```

Question 20

In the code to the right, what is output on line #1?

- A) 70
- B) 85
- C) 100
- D) 110
- E) 120

Question 21

In the code to the right, what is output on line #2?

- A) 132
- B) 137
- C) 146
- D) 148
- E) 157

Question 22

In the code to the right, what is output on line #3?

- A) 220
- B) 270
- C) 310
- D) 330
- E) 350

```
public static int dr(int A, int B)
{
    if (A < B)
        return dr(A+2,B+1) + A*B;
    return A*10;
}

// Client Code
out.print(dr(12,7)); // line #1
out.print(dr(4,6)); // line #2
out.print(dr(1,6)); // line #3
```

Question 23

What is the output of the code segment shown on the right?

- A) SITYO
- B) RSITY
- C) RSIT
- D) SITY
- E) ITYO

```
String St = "UNIVERSITYOFTEXAS";
int T = St.length();
St = St.substring(T/4,3*T/4);
St = St.substring(1,St.length()-1);
St = St.substring(1,St.length()-1);
out.print(St);
```

Question 24

Which could possibly be the output of the code segment shown on the right? Four options are not possible. Only one is possible.

- A) 25
- B) 75
- C) 125
- D) 175
- E) 225

```
int A = (int)(Math.random()*5 + 10);
int B = (int)(Math.random()*10 + 3);
int C = (int)(Math.random()*2 + 6);
out.print(A * C + B);
```

Question 25

In the code to the right, what is output on line #1?

- A) 46 B) 52 C) 62 D) 64 E) 68

Question 26

In the code to the right, what is output on line #2?

- A) 10 B) 14 C) 15 D) 16 E) 20

Question 27

In the code to the right, what is output on line #3?

- A) 5 B) 6 C) 7 D) 8 E) 9

```
public static int Red(int Number)
{
    Number = Number%10*10 + Number/10;
    return Number;
}

public static int White(int Number)
{
    Number = (int)(0.1* Number + 0.5)*10;
    return Number;
}

public static int Blue(int Number)
{
    Number = (int)(Math.sqrt(Number));
    return Number;
}

// Client Code
out.print(Red(64)); // Line #1
out.print(White(Blue(399))); // Line #2
out.print(Blue(Red(19))); // Line #3
```

Question 28

In the code to the right, what is output on line #1?

- A) 5 B) 6 C) 7 D) 8 E) 9

Question 29

In the code to the right, what is output on line #2?

- A) -192 B) =96 C) -48 D) -24 E) -12

Question 30

In the code to the right, what is output on line #3?

- A) -3 B) 6 C) 12 D) 24 E) 48

```
PriorityQueue<Integer> T;
T = new PriorityQueue<Integer>();
int x = -3;
while (x < 100)
{
    T.add(x);
    x = -2 * x;
}

out.println(T.size()); //Line #1
out.println(T.peek()); //Line #2
for(int y=1; y<=5; y++)
    T.remove();
out.println(T.peek()); //Line #3
```

Question 31

What is the output of the code segment shown on the right?

- A) 2 B) 4 C) 8 D) 16 E) 32

```
int A = 48;
int B = 32;
int C = 0;
for(int x = 1; x<=1000; x++)
    if (A%x==0 && B%x==0)
        C = x;
out.print(C);
```

Question 32

How many ones?

- A) 10 B) 11 C) 12 D) 13 E) 14

$$2^{15} = 32768$$

How many ones are in the binary representation of the base ten number 32764?

Question 33

What is the output of the code segment shown on the right?

- A) 120 B) 124 C) 128 D) 130 E) 136

```
out.print(Integer.parseInt("B4",12));
```

Question 34

In the client code to the right, what is output on line #1?

- A) 63
B) 56
C) 49
D) 42
E) 35

```
public class Football
{
    private int TD;
    private int FG;
    private int XP;

    public Football(int A, int B)
    {
        TD = A;
        FG = B;
        XP = A;
    }

    public int getScore()
    {
        if (TD > FG)
            return TD * 8 + XP;
        return FG * 5;
    }

    public int getFun()
    {
        if (TD != FG)
            return TD * 3 - FG;
        return XP * 10;
    }
}
```

Question 35

In the client code to the right, what is output on line #2?

- A) 8
B) 12
C) 20
D) 40
E) 48

```
//Client code
Football Bob = new Football(7,3);
Football Meg = new Football(4,4);
int A = Bob.getScore();
int B = Bob.getFun();
int C = Meg.getScore();
int D = Meg.getFun();
out.println(A);            //Line #1
out.println(D);            //Line #2
out.println(C + B);        //Line #3
```

Question 36

In the client code to the right, what is output on line #3?

- A) 38
B) 40
C) 42
D) 44
E) 46

Question 37

What is the output of the code segment shown on the right?

- A) 9 3
- B) 9 5
- C) 9 8
- D) 8 5
- E) 8 3

```
int R=8; int S=5; int T=9; int U=3;  
R = Math.max(R,S);  
S = Math.min(R,S);  
T = Math.min(U,T);  
U = Math.max(U,T);  
R = Math.max(R,T);  
S = Math.min(U,T);  
out.print(R + " " + S);
```

Question 38

What is the output of the code segment shown on the right?

- A) -4
- B) -2
- C) -1
- D) 1
- E) 2

```
String One = "Earl";  
String Two = "Campbell";  
out.print(One.compareTo(Two));
```

Question 39

Imagine you have an empty binary search tree. First, you add the five vowels into the tree, one at a time in this order (A,E,I,O,U). Next, you add to the tree the remaining consonants in alphabetical order, starting with B and ending with Z.

How many leafs would the tree have?

Write the number in blank #39.

Question 40

Evaluate the postfix expression to the right.

9 7 5 + 3 / 2 - 4 1 + + *

Write the number in blank #40.

★ANSWER KEY – CONFIDENTIAL★

UIL COMPUTER SCIENCE – 2024 District

Questions (+6 points for each correct answer, -2 points for each incorrect answer)

- | | | | |
|--------------|--------------|--------------|----------------|
| 1) D | 11) D | 21) C | 31) D |
| 2) A | 12) D | 22) D | 32) D |
| 3) D | 13) E | 23) D | 33) E |
| 4) A | 14) D | 24) B | 34) A |
| 5) A | 15) C | 25) A | 35) D |
| 6) B | 16) D | 26) E | 36) A |
| 7) B | 17) B | 27) E | 37) E |
| 8) D | 18) E | 28) C | 38) E |
| 9) D | 19) A | 29) A | *39) 5 |
| 10) C | 20) E | 30) D | *40) 63 |

* See "Explanation" section below for alternate, acceptable answers.

Note: Correct responses are based on **Java SE Development Kit 20 (JDK 20)** from Oracle, Inc. All provided code segments are intended to be syntactically correct, unless otherwise stated (e.g., "error" is an answer choice) and any necessary Java SE 20 Standard Packages have been imported. Ignore any typographical errors and assume any undefined variables are defined as used.

Explanations:

1.	D	<p>Change everything to base 8 first. 110011_2 becomes 63_8 and 64_{10} becomes 100_8. 63_8 times 100_8 is 6300_8.</p>
2.	A	<p>$12 - 27 + 17 \% 10 / 2$ $12 - 27 + 7 / 2$ $12 - 27 + 3$ $-15 + 3 = -12$</p>
3.	D	<pre>int A = 14; int B = 16; int C = 9; int D = 9; out.print(A % B); Prints 14 out.print(B % C); Prints 7 out.println(A / D); Prints 1 then carriage return (next line) out.print(C / D); Prints 1 out.println(D % C); Prints 0 then carriage return (next line) out.println(A + B / C + D); Prints 24 then carriage return (next line)</pre>
4.	A	<pre>String St3 = St2+St1.substring(0,3); St3 is "CITYKAN" out.print(St3.charAt(5)); Character #5 is the A</pre>
5.	A	<pre>boolean A = true; A is true boolean B = !A; B is false boolean C = A B; C is true (Or needs at least 1 true) boolean D = B && C; D is false (And need 2 trues) boolean E = A ^ D; E is true (Xor needs exactly 1 true) out.print(E);</pre>
6.	B	<p>This adds the numbers 5, 7, and 9 to get 21.</p>
7.	B	<pre>double A = Math.floor(8.7); A = 8.0 double B = Math.ceil(A - 0.1); B = 8 double C = Math.sqrt(A + B); C = 4 int D = (int) C; D = 4 out.print(D);</pre>
8.	D	<pre>int A = 5; A=5 int B = A - 1; B=4 int C = B + 2; C=6 if (A > B) true C++; C=7 if (C < A) false B++; if (B < C) true A++; A=6 out.print(A + B + C); 17</pre>

9.	D	<pre>int[] Stuff = {4,2,1,0,3}; int A = Stuff[Stuff[2]]; int B = Stuff[Stuff[3]]; int C = Stuff[Stuff[0]]; out.print(A + B + C);</pre>	<pre>A= Stuff[1] = 2 B= Stuff[0] = 4 C= Stuff[4] = 3 9</pre>
10.	C	<pre>elf[10] = 10 * 2 + 5 = 25 elf[20] = 20 * 2 + 5 = 45 70 is printed</pre>	
11.	D	<pre>Fred starts at [7,5,8,3,9,1,4,0,2] Fred.set(5,2); [7,5,8,3,9,2,4,0,2] Fred.add(4,6); [7,5,8,3,6,9,2,4,0,2] for (int x=1; x<=5; x++) Fred.remove(1); Remove the 5,8,3,6,9 out.print(Fred); [7,2,4,0,2]</pre>	
12.	D	<pre>for(int i=-5, T=1; i<=5; i=i+3, T++) out.print(i*T + " "); Pass #1 i = -5 T=1 i*T = -5 Pass #2 i = -2 T=2 i*T = -4 Pass #3 i = 1 T=3 i*T = 3 Pass #4 i = 4 T=4 i*T = 16</pre>	
13.	E	<pre>6 + 10 & 4 + 10 << 2; 16 & 4 + 10 << 2 16 & 14 << 2 16 & 56 10000 & 111000 = 0010000 = 16</pre>	
14.	D	<pre>out.print(Long.SIZE/Short.SIZE); out.print(Integer.SIZE/Byte.SIZE);</pre>	<pre>64/16 = 4 32/8 = 4</pre>
15.	C	<pre>Heat = "5 1 2 9 2 6 7 4 1 7"; int T = Sc.nextInt(); for (int x=0; x<=T; x++) Sc.next(); T += Sc.nextInt() + Sc.nextInt(); out.print(T);</pre>	<pre>T=5 skip 6 1,2,9,2,6,7 Add 4 and 1 to T</pre>
16.	D	<pre>int G = 0; for(int A=-3; A<=3; A++) for(int B=0; B<=100; B+=10) G++; out.print(G);</pre>	<pre>7 iterations 11 iterations 77</pre>

17.	B	String Plasma = "QWERTYUIOP"; Traverses entire string. If letter X is before letter X+1 in the alphabet, it adds 5 to the total. QW, ER, RT, TY, IO, OP = 6 times 5 = 30
18.	E	int A = 20; A is 20 int B = (A>16) ? 21:15; B is 21 since A > 16 int C = (A>B) ? 7:12; C is 12 since is not greater than B out.print(A + B + C); 53
19.	A	int[]Speed = {4,1,4,1,4,1}; int[]Velocity = {5,10,15,20,25,30,35,40}; int Texas = Speed[0]; for(int power: Speed) Texas += Velocity[power]; out.print(Texas); Texas = 4 + Velocity[4] + Velocity[1] + Velocity[4] + Velocity[1] + Velocity[4] + Velocity[1] Texas = 4 + 25 + 10 + 25 + 10 + 25 + 10 = 109
20.	E	public static int dr(int A, int B) { if (A < B) return dr(A+2,B+1) + A*B; return A*10; } dr(12,7) = 12*10 = 120
21.	C	dr(4,6) = dr(6,7) + 24 = 122 + 24 = 146 dr(6,7) = dr(8,8) + 42 = 80 + 42 = 122 dr(8,8) = 80
22.	D	dr(1,6) = dr(3,7) + 6 = 324 + 6 = 330 dr(3,7) = dr(5,8) + 21 = 303 + 21 = 324 dr(5,8) = dr(7,9) + 40 = 263 + 40 = 303 dr(7,9) = dr(9,10) + 63 = 200 + 63 = 263 dr(9,10) = dr(11,11) + 90 = 110 + 90 = 200 dr(11,11) = 110
23.	D	String St = "UNIVERSITYOFTEXAS"; int T = St.length(); T=17 St = St.substring(T/4, 3*T/4); St.substring(4,12) = "ERSITYOF" St = St.substring(1, St.length()-1); "RSITYO" St = St.substring(1, St.length()-1); "SITY"
24.	B	A = (int) (Math.random()*5 + 10); A= [10,14] B = (int) (Math.random()*10 + 3); B= [3,12] C = (int) (Math.random()*2 + 6); C= [6,7] Substituting minimum values, we get 10*6 + 3 = 63 Substituting maximum values, we get 14*7 + 12 = 110 Thus only values [63,110] may be printed.

25.	A	Red takes a two-digit number and reverses the digits.
26.	E	White rounds a number to the nearest 10.
27.	E	Blue, takes a square root and chops off the decimal. <code>out.print (Red(64));</code> reverse digits = 46 <code>out.print (White (Blue(399)));</code> square root chopped is 19, nearest 10 is 20 <code>out.print (Blue (Red(19)));</code> reverse digits is 91, square root chopped is 9
28.	C	<code>PriorityQueue<Integer> T;</code> <code>T = new PriorityQueue<Integer>();</code> The numbers -3, 6, -12, 24, -48, 96, and -192 are added to a T. The size is 7. <code>out.println(T.peek());</code> //Line #2 <code>for(int y=1; y<=5; y++)</code> <code>T.remove();</code> <code>out.println(T.peek());</code> //Line #3
29.	A	When we peek, we see the item in the front - always the smallest item. We see -192
30.	D	The loop zaps the first item 5 times - always the smallest of the remaining items. Thus -192, -48, -12, -3, and 6 are removed. Now, when we peek, we see the 24.
31.	D	Although the loop iterates a ridiculous number of times, this finds the greatest common factor of 48 and 32 ... 16
32.	D	16 or 2^4 is 10000 in binary. Subtract 4 and you get 1100 (2 ones) 32 or 2^5 is 100000 in binary. Subtract 4 and you get 11100 (3 ones) 64 or 2^6 is 1000000 in binary. Subtract 4 and you get 111100 (4 ones) 2^{15} minus 4 should have 13 ones.
33.	E	$B_{4_{12}} = 136_{10}$
34.	A	Since $7 > 3$ Bob.getScore() will be $7*8+7 = 63$
35.	D	Since $4 == 4$ Meg.getFun() will be $4*10 = 40$
36.	A	Since $7 != 3$ Bob.getFun() will be $7*3-3 = 18$ Since 4 is not greater than 4 Meg.getScore() will be $4*5 = 20$ $18+20 = 38$
37.	E	<code>int R=8; int S=5; int T=9; int U=3;</code> <code>R = Math.max(R,S);</code> R=8 <code>S = Math.min(R,S);</code> S=5 <code>T = Math.min(U,T);</code> T=3 <code>U = Math.max(U,T);</code> U=3 <code>R = Math.max(R,T);</code> R=8 <code>S = Math.min(U,T);</code> S=3 <code>out.print(R + " " + S);</code> 8 3
38.	E	"Earl".compareTo("Campbell") focuses on the ASCII values if the E and the C. E is two greater than C
39.	5	D, H, T, N, and Z are all leafs (leaves?)
40.	63	$9 \underline{75} + 3 / 2 - 4 \ 1 + + *$ $9 \underline{123} / 2 - 4 \ 1 + + *$ $9 \underline{42} - 4 \ 1 + + *$ $9 \underline{241} + + *$ $9 \underline{25} + *$ $9 \underline{7} * = 63$