

UIL COMPUTER SCIENCE WRITTEN TEST

2023 REGION

APRIL 2023

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 – 2023 REGION

Note: Correct responses are based on **Java SE Development Kit 17 (JDK 17)** 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 17 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

Which of the below has a value different from the other four?

- A) 10110101_2 B) 2311_4 C) 265_8 D) 183_{10} E) $B5_{16}$

Question 2

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

- A) 8 B) 1066 C) 52 D) 53 E) 9

```
out.print(800 / 3 % 10 + 800 / 10 % 3);
```

Question 3

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

- A) `\12.121212121212\\
nn12`
B) `\12.121212120000\\
nn12`
C) `12.12\\nnn12`
D) `\12.12\\
nn12`
E) `\12.121212120000\\nn12`

```
double March = 12.12121212;  
String St = "\\%.12f\\\\" + nnn + "%d";  
out.printf(St, March, (int)March);
```

Question 4

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

- A) Com B) omp C) cie D) nce E) ute

```
String Go = "ComputerScience";  
String Fight = Go.substring(4);  
String Win = Fight.substring(4);  
out.print(Win.substring(1,4));
```

Question 5

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

- A) true
B) false

```
boolean R = !false;  
boolean S = R || false;  
boolean T = R && S;  
boolean U = S ^ T;  
out.print(U);
```

Question 6

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

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

```
double M = Math.sqrt(80);  
out.print((int)Math.cbrt(M*M*M));
```

Question 7

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

- A) 5 B) 45 C) 50 D) 60 E) 75

```
int U = 240 % 100;  
int I = U % 25;  
int L = I % 10;  
out.print(U + I + L);
```

Question 8

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

- A) 10 2 4
- B) 11 10 11
- C) 10 10 10
- D) 11 10 10
- E) 10 10 11

```
int D = 10;
int R = 2;
int P = 4;
if (D + R > P)
    P = D;
else
    D = P;
if (D + P > R)
    R = D;
else
    D = R;
if (P + R > D)
    D++;
else
    D--;
out.print(D + " " + P + " " + R);
```

Question 9

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

- A) 125
- B) 12514
- C) 14
- D) 5
- E) 8

```
String St = "";
for(int x = 1; x < 10; x = x * 3 - 1)
    St += x;
out.println(St);
```

Question 10

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

- A) 32 B) 34 C) 35 D) 36 E) 38

```
int[] perfect = new int[25];
for(int x=0; x<24; x++)
    perfect[x] = x * x - 1;
out.print(perfect[10] - perfect[8]);
```

Question 11

What is output by the code segment to the right?

- A) 2 B) 6 C) 12 D) 20 E) 28

```
String St = "Z 2 X 4 C 6 V 8 B 10";
Scanner B = new Scanner(St);
int Hello = 0;
for(int x = 1; x <= 4; x++)
{
    B.next();
    Hello += B.nextInt();
}
out.print(Hello);
```

Question 12

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

- A) 50 B) 25 C) 4 D) 33 E) 17

```
int strange = 100;
for(int x = 2; x <= 4; x++)
    strange /= x;
out.print(strange);
```

Question 13

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

- A) 44
- B) 321
- C) 41
- D) 322
- E) 324

```
int Q = 20;
out.print(Q << 2 + 2 ^ 3 >> 1);
```

<p>Question 14</p> <p>What is the output of the code segment shown on the right?</p> <p>A) 18 B) 16 C) 17 D) 14 E) 15</p>	<pre>out.println(Byte.MAX_VALUE/Byte.SIZE);</pre>
<p>Question 15</p> <p>What is output by the code segment to the right?</p> <p>A) [10, 30, 50, 70, 90] B) [60, 70, 80, 90, 100] C) [20, 40, 60, 80, 100] D) [20, 30, 40, 50, 60] E) [10, 20, 40, 50, 70]</p>	<pre>ArrayList<Integer> keys; keys = new ArrayList<Integer>(); for(int x = 10; x<=100; x+=10) keys.add(x); for(int x = 1; x<= 5; x++) if (keys.get(x)%10 == 0) keys.remove(x); out.print(keys);</pre>
<p>Question 16</p> <p>What is the output of the code segment shown on the right?</p> <p>A) 60 B) 72 C) 66 D) 55 E) -66</p>	<pre>int z = 0; for(int x=1, y=-5; x<=12; x++, y++) z = x * y; out.print(z);</pre>
<p>Question 17</p> <p>What is the output of the code segment shown on the right?</p> <p>A) 60 B) 50 C) 70 D) 40 E) 90</p>	<pre>int N = 100; int A = 0; int B = 0; int C = 0; for(int x=1; x<=N; x++) { if (x%5==0) A++; if (x%2==0) B++; if (x%5==0 && x%2==0) C++; } System.out.print(A + B - C);</pre>
<p>Question 18</p> <p>What is the output of the code segment shown on the right?</p> <p>A) 2.0 B) 2 C) 1.0 D) 1 E) 0</p>	<pre>out.print(Math.pow(64 ,1/6));</pre>
<p>Question 19</p> <p>What is the output of the code segment shown on the right?</p> <p>A) BEMJALSAM B) BESOEMSAL C) BOBJENSAM D) BSJEAANNL E) BEMJAMSAL</p>	<pre>String[]L = {"BOB", "JEN", "SAM", "PAM", "MEL"}; String St = ""; for(int x=1;x<=3;x++) { St += L[x-1].substring(0,1); St += L[x].substring(1,2); St += L[x+1].substring(2,3); } out.print(St);</pre>

Question 20

In the code segment to the right, what is the output of line 1?

- A) 8 B) 7 C) 0 D) 10 E) 9

Question 21

In the code segment to the right, what is the output of line 2?

- A) 8 B) 7 C) 0 D) 10 E) 9

Question 22

In the code segment to the right, what is the output of line 3?

- A) 8 B) 7 C) 0 D) 10 E) 9

```
public class Alpha
{
    private int A;

    public Alpha(int AA)
    {
        A = AA;
    }

    public int getA()
    {
        return A + 1;
    }
}

public class Beta extends Alpha
{
    private int A;

    public Beta(int AA)
    {
        super(AA + 5);
    }

    public int getA()
    {
        return A;
    }
}

////////////////////////////////////
// Client code
Alpha Amp = new Alpha(7);
Beta Bob = new Beta(10);
Alpha Art = new Alpha(Amp.getA());
System.out.println(Amp.getA()); // Line 1
System.out.println(Bob.getA()); // Line 2
System.out.println(Art.getA()); // Line 3
```

Question 23

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

- A) 1 B) 16 C) 31 D) 32 E) 63

```
out.print(31 ^ 32);
```

Question 24

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

- A) 64 B) 32 C) 0 D) -32 E) -64

```
out.print('E' - 'T' + 't' - 'e');
```

Question 25

In the code segment to the right, what is the output of line 1?

- A) 1 B) 2 C) 4 D) 7 E) 9

```
PriorityQueue<Integer>PQ;
PQ = new PriorityQueue();
```

Question 26

In the code segment to the right, what is the output of line 2?

- A) 1 B) 2 C) 4 D) 7 E) 9

```
int[]List = {5,1,2,9,2,6,7,4,1,7};
for(int x=0; x<List.length; x++)
    PQ.add(List[x]);
```

```
out.println(PQ.remove()); // line #1
PQ.remove();
out.println(PQ.peek());   // line #2
out.println(PQ);         // line #3
```

Question 27

In the code segment to the right, what is the output of line 3?

- A) [2, 9, 2, 6, 7, 4, 1, 7]
 B) [2, 2, 4, 5, 6, 7, 7, 9]
 C) [9, 7, 7, 6, 5, 4, 2, 2]
 D) [2, 4, 2, 7, 5, 6, 7, 9]
 E) [4, 5, 6, 7, 7, 9]

Question 28

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

- A) E B) N C) M D) C E) S

```
Stack<Character>Bunch;
Bunch = new Stack<Character>();
```

Question 29

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

- A) E B) N C) M D) C E) S

```
String St = "COMPUTERSCIENCE";
for (int x=0; x<St.length(); x++)
    if(!Bunch.contains(St.charAt(x)))
        Bunch.push(St.charAt(x));
```

Question 30

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

- A) 6 B) 5 C) 4 D) 3 E) 2

```
System.out.print(Bunch.peek()); //Line 1
for (int x=1; x<=8; x++)
    Bunch.pop();
System.out.print(Bunch.pop()); //Line 2
System.out.print(Bunch.size()); //Line 3
```

Question 31

Solve the problem stated to the right.

- A) 16 seconds
 B) 32 seconds
 C) 64 seconds
 D) 409 seconds
 E) 1024 seconds

The Big O Notation for a sorting routine is $O(n * \log_2 n)$. When we sort a list of 10,000 elements, the process takes 0.4 seconds. How long do we predict the same sort will work on a list of 320,000 numbers?

Question 32

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

- A) 20
- B) 40
- C) 25
- D) 30
- E) 10

```

TreeMap<Integer,Integer>Oak;
Oak= new TreeMap<Integer,Integer>();
for (int x=1; x<=10; x++)
    Oak.put(x,x*2);
for (int y=10; y>=1; y-=2)
    for (int x=20;x<=21;x++)
        Oak.put(y,x);
out.println(Oak.size()); // Line #1
out.println(Oak.get(10)); // Line #2
out.println(Oak.get(3)); // Line #3

```

Question 33

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

- A) 10
- B) 5
- C) 20
- D) 21
- E) 6

Question 34

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

- A) 7
- B) 21
- C) 20
- D) 3
- E) 6

Question 35

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

- A) 1 B) 4 C) 5 D) 6 E) 10

```

int[][]BP = new int[8][8];
BP[1][1] = 1;
for (int y = 2; y<=7; y++)
    for (int x = 1; x<=7; x++)
        BP[y][x]= BP[y-1][x]+BP[y-1][x-1];
out.print(BP[6][5]);

```

Question 36

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

- A) 7 B) 8 C) 24 D) 22 E) 23

```

int A = 16;
A += Math.sqrt(A-1);
A += Math.sqrt(A+1);
out.print(A);

```

Question 37

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

- A) 1
- B) 2
- C) 3
- D) 4
- E) 5

```

int C=0;
String A = "R2-D2";
if(A.matches(".....")) C++;
if(A.matches(".2.2")) C++;
if(A.matches("[A-T]*")) C++;
if(A.matches("2.*")) C++;
if(A.matches(".*2")) C++;
if(A.matches(".*[0-9].*[0-9]")) C++;

out.print(C);

```

Question 38

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

- A) ABCDE
- B) EDCBA
- C) ABCDEEDCBA
- D) ABCDEDCBA
- E) AABBCCDDEE

```
String One = "ABCDE";  
String Two = "";  
for(int x=0; x<One.length(); x++)  
    Two = One.substring(x,x+1) + Two;  
out.print(Two);
```

Question 39

There are 16 different combinations of A, B, C, and D in the code on the right. How many combinations would result in the value "true" being printed?

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

Question 40

Consider a binary tree with 1023 nodes. What is the greatest number of leaves that the tree could contain?

★ ANSWER KEY – CONFIDENTIAL ★

UIL COMPUTER SCIENCE – 2023 REGION

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

1) D	11) D	21) C	31) C
2) A	12) C	22) E	32) E
3) B	13) B	23) E	33) D
4) C	14) E	24) C	34) E
5) B	15) A	25) A	35) C
6) B	16) B	26) B	36) E
7) D	17) A	27) D	37) C
8) D	18) C	28) B	38) B
9) A	19) E	29) C	*39) 10
10) D	20) A	30) E	*40) 512

KEY

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

Note: Correct responses are based on **Java SE Development Kit 17 (JDK 17)** 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 17 Standard Packages have been imported. Ignore any typographical errors and assume any undefined variables are defined as used.

Explanations:

1.	D	<p>Convert the 4 powers-of-two numbers to base 16. $10110101_2 = 1011\ 0101_2$ which is $B5_{16}$ $2311_4 = 10\ 11\ 01\ 01_2 = 1011\ 0101_2$ which also is $B5_{16}$ $265_8 = 010\ 110\ 101_2 = 1011\ 0101_2$ which also is $B5_{16}$ Well, $B5_{16}$ is $B5_{16}$ $B5_{16} = 11*16 + 5*1 = 181_{10}$ So, the odd one out is 183_{10}</p>
2.	A	<p>Use order of operations. $800 / 3 \% 10 + 800 / 10 \% 3$ $266 \% 10 + 800 / 10 \% 3$ $6 + 800 / 10 \% 3$ $6 + 80 \% 3$ $6 + 2 = 8$</p>
3.	B	<p>You will be printing 12.12121212 and 12 with the format defined by String St. <code>\\</code> gives us one \ <code>%.12f</code> prints the double with 12 decimal places. That gives us 12.121212120000 <code>\\\</code> produces \<\ <code>\n</code> provides a new line. <code>nn</code> gives us, well nn <code>%d</code> formats 12 Thus... <code>\12.121212120000\<\ nn12</code></p>
4.	C	<p><code>Go = "ComputerScience"</code> <code>Fight = "uterScience"</code> <code>Win = "Science"</code> It prints "cie"</p>
5.	B	<p><code>R = true (!false)</code> <code>S = true (true false)</code> <code>T = true (true && true)</code> <code>U = false (true ^ true)</code></p>
6.	B	<p><code>M = 8.94</code> <code>M*M*M</code> will be a number greater than 512 but less than 729 So the cube root is greater than 8 but less than 9. <code>(int)</code> casts the number to 8</p>
7.	D	<p><code>U = 40 (240%100)</code> <code>I = 15 (40 % 25)</code> <code>L = 5 (15 % 10)</code> <code>40+15+5 = 60</code></p>
8.	D	<p><code>D=10 R=2 P=4</code> <code>P</code> becomes 10 since <code>10+2>4</code> <code>R</code> becomes 10 since <code>10+10>2</code> <code>D</code> becomes 11 since <code>10+10>10</code> It prints 11 10 10</p>
9.	A	<p>The loop goes through the values 1, 2, and 5 before 14 stops the process. Since <code>St</code> is a String, the values are concatenated onto the <code>St</code> producing "125"</p>
10.	D	<p><code>perfect[10]</code> is 99 ($10*10-1$) <code>perfect[8]</code> is 63 ($8*8-1$) $99-63 = 36$</p>
11.	D	<p><code>"Z 2 X 4 C 6 V 8 B 10"</code> The loop iterates 4 times. Each time, it reads the String first and does nothing with it. It then reads the integer and adds it to Hello $2+4+6+8 = 20$</p>

12.	C	This divides 100 by 2 to get 50 Then it divides 50 by 3 to get 16 Then it divides 16 by 4 to get 4
13.	B	Follow the order of precedence... <u>Q</u> << 2 + 2 ^ 3 >> 1 20 << <u>2 + 2</u> ^ 3 >> 1 <u>20</u> << 4 ^ 3 >> 1 320 ^ <u>3</u> >> 1 320 ^ 1 = 101000000 ^ 000000001 = 101000001 = 321
14.	E	Byte.MAX_VALUE = 127 Byte.SIZE = 8 127/8 = 15
15.	A	After the first loop, keys begins with [10,20,30,40,50,60,70,80,90,100] In the second loop, we begin at position 1 and delete items that are evenly divisible by 10 (all of the numbers in the list are evenly divisible by 10). But, when we delete the 20, the items move back one position, so we miss the 20 and delete the 30 on the next step. This process continues. [10,30,50,70,90]
16.	B	In the loop, x begins at 1 and does 12 iterations reaching 12. As x does its work, y starts at -5 and does the same 12 iterations stopping at 6. The final printed z is the final x*y = 72
17.	A	A counts the multiples of 5 in the range [1,100] which is 20. B counts the multiples of 2 in the range [1,100] which is 50. C counts the multiples of 10 in the range [1,100] which is 10. 20+50-10 = 60
18.	C	1/6 = 0 64 to the 0 power is 1. Math.pow() returns a double. The answer is 1.0
19.	E	In the first iteration of the loop, St = "B" St = "BE" St = "BEM" In the next iteration of the loop, St = "BEMJ" St = "BEMJA" St = "BEMJAM" In the next iteration of the loop, St = "BEMJAMS" St = "BEMJAMSA" St = "BEMJAMSAL"
20.	A	Amp's A attribute is equal to 7. Alpha's getA() returns A+1 8
21.	C	Bob's A attribute never changes from the default zero. When the constructor is called, it sends the argument 15 up to Alpha and its constructor. So, when we call Bob.getA() it returns a 0.
22.	E	Art takes Amp.getA() which is 8, and sends it to the constructor to be used. Art's A attribute is indeed 8, but Alpha's getA() returns A+1. 9
23.	E	31 ^ 32 011111 ^ 100000 = 111111 = 63 in base 10
24.	C	If you know your ASCII values for the letters ('A' is 65 and 'a' is 97) you can do the arithmetic, but... The difference between 'E' and 'T' will be the same as 't' and 'e' with one being positive and the other negative. So, the answer is 0.
25.	A	A PriorityQueue is a Min-Heap, so when the first item is removed, it will be the smallest value in the list. That value is 1
26.	B	Two items are removed from the Priority Queue. So, the two 1's are gone. When we peek(), we see the new smallest item, 2.

27.	D	<p>After two removals, we now want to see what the list looks like. It is important to learn the algorithm used when deleting from a Priority Queue. Here is the evolution of the list. [1, 1, 2, 2, 5, 6, 7, 9, 4, 7] - This is the original Priority Queue. [1, 2, 2, 4, 5, 6, 7, 9, 7] - The Priority Queue after the first item is removed [2, 4, 2, 7, 5, 6, 7, 9] - The Priority Queue after the second item is removed</p> <p>But in desperation, there are ways to narrow it down. The first element must be the smallest, so C and E are out. Choice A would be nearly impossible, because at this point the largest value, 9, should not be on row one of the tree. Choice B is possible, but very unlikely, since the list of 8 items is sorted. D is the answer and would have been the best guess. But, learn the algorithm. It is very cool and will take you only ten minutes of practice.</p>
28.	B	<p>In this problem, the only letters allowed to be pushed on the Stack are items not already on the Stack. It allows no duplicates. So, the initial Stack from bottom to top is: C O M P U T E R S I N. When we peek(), we see the N.</p>
29.	C	<p>The Stack is C O M P U T E R S I N The loop pops the 8 items on the top, leaving us with C O M. When we pop() the next item, it is M.</p>
30.	E	<p>After all of this, the Stack is C O. The size() is 2</p>
31.	C	<p>$O(n * \log_2 n)$ This is the Big O notation for routines like the Quick Sort. It generally involves routines where there is a linear iteration and within it a process of cutting things in half. In this problem, we must have to do the arithmetic. $n = 32$ This is determined by seeing how many times larger the second list is than the original list. Plugging into the formula: $32 * \log_2 32$ $\log_2 32 = 5$ $32 * \log_2 32 = 160$ Now take 160 times the original time 0.4 and we get 64.</p>
32.	E	<p>The first loop creates the following relationships. 1 -> 2 2 -> 4 3 -> 6 4 -> 8 5 -> 10 6 -> 12 7 -> 14 8 -> 16 9 -> 18 10 -> 20 The nested loops reassign 10 of the mappings first to values of 20, then to values of 21. 1 -> 2 2 -> 21 3 -> 6 4 -> 21 5 -> 10 6 -> 21 7 -> 14 8 -> 21 9 -> 18 10 -> 21 The size is 10</p>
33.	D	<p>Looking at the explanation of #32, Oak.get(10) = 21</p>
34.	E	<p>Looking at the explanation of #32, Oak.get(3) = 6</p>

35.	C	<p>This algorithm generates the values of Pascal's triangle. Formatting would print a prettier picture, but we are focusing on Row 6</p> <pre> 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 2 1 0 0 0 0 0 1 3 3 1 0 0 0 0 1 4 6 4 1 0 0 0 1 5 10 10 5 1 0 0 1 6 15 20 15 6 1 </pre> <p>Row 6 0 1 5 10 10 5 1 0 We want element 5 in that row = 5</p>
36.	E	<p>A = 16 Math.sqrt(15) is a little less than 4. += type casts the expression on the right. We add 3 to A making it 19. Math.sqrt(20) is more than 4, but less than 5.. Again with the type casting. We add 4 more to A to give us 23.</p>
37.	C	The 1st, 5th, and 6th if statements evaluate to true.
38.	B	<p>This is a classic method of reversing a String. Two = Two + One.substring(x,x+1) would produce the original String. But, reversing the two expressions on the right gives us a reversal. Two = One.substring(x,x+1) + Two</p>
39.	10	<p>Here is the truth table for the problem: The order of operations will be ^ && </p> <pre> 0000 = 0 0001 = 1 0010 = 0 0011 = 1 0100 = 0 0101 = 1 0110 = 0 0111 = 1 1000 = 0 1001 = 1 1010 = 1 1011 = 1 1100 = 1 1101 = 1 1110 = 0 1111 = 1 </pre> <p>If D is true, the expression is true because of the placement of the "or." That gives us 8 answers. If D is false, the expression is true if A is true and B and C are opposites. That gives us 1010 and 1100. So, there are 10 combinations that will make the expression true.</p>
40.	512	The perfect scenario for most leaves (leafs?) would be a perfectly balanced 10 level binary tree with 511 interior nodes and 512 nodes on the bottommost level. All 512 would be leaves.