UIL COMPUTER SCIENCE WRITTEN TEST

2018 REGION

APRIL 2018

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)
```

UIL COMPUTER SCIENCE WRITTEN TEST – 2018 REGION

Note: Correct responses are based on **Java SE Development Kit 8 (JDK 8)** 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 8 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 following is not equal to $AF_{16} + 77_8$?

A) 22201_3 **B)** EE_{16} **C)** 238_{10} **D)** 356_8 **E)** 11101110_2

Question 2.

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

A) −1 **B)** 1 **C)** 5 **D)** −5 **E)** 0

```
out.println(-8%3+15%-4);
```

Question 3.

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

A) %,-12345.33

B) (12,345.33)

C) -12345.3

D) -12345.33

E) -12,345.33

```
out.printf("%,7.2f",-12345.329);
```

Question 4.

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

A) 18 **B)** 19 **C)** 20

D) 21 **E)** -1

```
String s="thirty seconds to mars";
int x=s.indexOf(s.substring(18, 19),20);
out.print(x);
```

Question 5.

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

A) true **B)** false

```
boolean b=false;
out.print(b||b&&b^b);
```

Question 6.

Which of the following may replace **<code>** in the line of code shown on the right to ensure that it will compile?

A) int **B)** float **C)** double **D)** long

E) More than one of the above.

```
<code> x=Math.round(Math.PI);
```

Question 7.

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

A) m **B)** n **C)** o **D)** p **E)** q

```
double d='c'+12.75;
out.print((char)d);
```

Question 8.

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

A) 32

B) 43

C) 33

D) 38

E) 37

```
int n=1593535253;
int s=0;
while(n>0) {
  switch(n%10) {
  case 3:s++;break;
  case 5:s+=n%10;break;
  default:s+=5;
  }
  n/=10;
}
out.print(s);
```

Question 9.

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

A) `1 2 3 4`

B) `0 1 2 3 4`

C) `1 2 3 4 5`

D) `0 1 2 3 4 5`

E) `0 2 4`

```
int k=0;
while(++k<5)
     out.print(k+" ");
```

Question 10.

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

A) `[6, 7, 8, 12, 10]`
`[1, 11, 3, 4, 5]`

B) `[1, 11, 3, 12, 5]`
`[1, 11, 3, 12, 5]`

C) `[6, 11, 8, 12, 10]`
`[6, 11, 8, 12, 10]`

D) `[1, 11, 3, 4, 5]`
`[6, 7, 8, 12, 10]`

E) Error. Throws ArrayIndexOutOfBoundsException.

```
int a[]= {1,2,3,4,5};
int b[]= {6,7,8,9,10};
b=a;
a=b;
b[1]=11;a[3]=12;
out.println(Arrays.toString(a));
out.println(Arrays.toString(b));
```

Question 11.

Which of the following may not replace **<code>** in the code segment shown on the right?

A) `nextInt()`

B) `nextDouble()`

C) `next()`

D) `nextFloat()`

E) None of the above. All may replace **<code>**.

```
Scanner s=new Scanner("2 12 6 8 1");
double sum=0;
while(s.hasNext())
     sum+=s.<code>;
out.print(sum);
```

Question 12.

Which of the following code segments will find and print the sum of the odd numbers within the array `a`? Array `a` has been initialized with this line : `int[] a= {5,6,7,4,1,9,8};`

A.

```
int sum=0;
int x=0,y=a.length;
while(x<=y) {
  if(a[x]%2!=0)
    sum+=a[x];
  x++;
  }
out.print(sum);
```

B.

```
int sum=0;
for(int x=0;x<a.length;x++)
  if(a[x]%2==0)
    sum+=a[x];
out.print(sum);
```

C.

```
int sum=0;
for(int x=1;x<=a.length;x++)
  if(a[x]%2!=0)
    sum+=a[x];
out.print(sum);
```

D.

```
int sum=0;
int x=0,y=a.length;
while(x<y)
  if(a[x]/2!=0)
    sum+=a[x];
  x++;
out.print(sum);
```

E.

```
int sum=0;
for(int x=0;x<a.length;x++)
  if(a[x]%2!=0)
    sum+=a[x];
out.print(sum);
```

Question 13.

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

A) 7

B) 21

C) 4

D) 0

E) Error. The operator | is undefined for the int data type.

```
int g=5,h=2,i=3;
out.print(g|h*i);
```

Question 14.

What is printed by the segment of code shown on the right?

A) 2147450880

B) 2147450881

C) -32895

D) 32895

E) 32640

```
int i=Short.MAX_VALUE-Byte.MIN_VALUE;
out.print(i);
```

Question 15.

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

A) [5, 6, 4, 8]

B) [5, 6]

C) [5, 6, 4]

D) [5, 6, 8]

E) Error. Throws an IndexOutOfBoundsException .

```
ArrayList<Integer> a=new
ArrayList<Integer>();
a.add(5);
a.add(3);
a.add(8);
a.add(1, 6);
a.set(2, 4);
a.remove(3);
out.print(a);
```

Question 16.

What is the output of the client code shown to the right?

A) gouiluilill

B) liluilgouil

C) ggogouil

D) gouilgog

E) ggogougouigouil

```
public static String rec(String s) {
  if(s.length()==1)
    return s;
  else
    return rec(s.substring(s.length()/2))+s;
}

//client code
String s="gouil";
out.print(rec(s));
```

Question 17.

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

A) [8, 9, 12]
[3, 4, 11]
[9, 10, 11]

B) [8, 2, 12]
[9, 3, 13]
[10, 6, 14]

C) [8, 9, 10]
[3, 4, 5]
[9, 10, 11]

D) [8, 2, 12]
[9, 3, 13]
[10, 4, 14]

E) There is no output due to an error.

```
int[][] mat= {{8,2,12},
              {3,1,11},
              {9,6,0}};
for(int x=0;x<mat.length;x++)
  for(int y=1;y<mat[x].length;y++)
    if(mat[x][y]<=mat[x][y-1])
      mat[x][y]=mat[x][y-1]+1;
for(int[] m:mat)
  out.println(Arrays.toString(m));
```

Question 18.

The line shown on the right is an example of _____.

A) scope restriction

B) parsing

C) coupling

D) casting

E) auto-boxing

```
Integer i=35;
```

Question 19.

Which of the following must replace **<code>** in class `Cls`?

A) `extends`

B) `implements`

C) `inherits`

D) `overrides`

E) `overloads`

Question 20.

Which of the methods shown in class `Cls` are required?

A) `m1`

B) `m2`

C) `m3`

D) All of the above.

E) None of the above.

Question 21.

All of the methods shown within the `Intf` interface are _____?

A) static

B) comparable

C) abstract

D) extendable

E) More than one of the above.

Question 22.

Assuming that **<code>** has been correctly replaced, what is the output of the client code shown on the right?

A) `truetrue`

B) `truefalse`

C) `falsetrue`

D) `falsefalse`

E) There is no output due to an error.

```
//Use the following code to answer questions
//19 – 22.

public interface Intf {
      public String m1(String s);
      public boolean m2(String s,int i);
      public int m3(String s,char c);
}

public class Cls <code> Intf {

      public String m1(String s) {
            return null;
      }

      public boolean m2(String s, int i) {
            return false;
      }

      public int m3(String s, char c) {
            return 0;
      }
}

//Client code
Intf i=new Cls();
Cls c=new Cls();
out.print(i instanceof Cls);
out.print(c instanceof Intf);
```

Question 23.

What is printed by the code shown on the right?

A) 01110

B) 11100

C) 111

D) 1

E) 10000

```
int x=7<<2;
out.print(Integer.toBinaryString(x));
```

Question 24.

The method `sort` shown on the right is intended to implement the Quicksort algorithm. Which of the following should replace **<code 1>** so that the middle element in the partition is assigned to `z`?

A) `list[(right-left)/2]`

B) `list[list.length/2]`

C) `left+right/2`

D) `list[(left+right)]`

E) `list[(left+right)/2]`

Question 25.

What is the purpose of **line #1**?

A) Iterate until a value less than the pivot value is found.

B) Iterate until a value greater than the pivot value is found.

C) Change the pivot value to be the element stored in `list[x]`.

D) Find the smallest value in the left hand partition of `list`.

E) Find the largest value in the right hand partition of the `list`.

Question 26.

Which of the following lines of code must replace **<code 2>** and **<code 3>** so that sort will compile and execute correctly?

A)
```
sort(list,y,left);
sort(list,x,right);
```

B)
```
sort(list,left,y);
sort(list,right,x);
```

C)
```
sort(list,left,y);
sort(list,x,right);
```

D)
```
sort(list,left,right);
sort(list,x,y);
```

E)
```
sort(list,left,right);
//Only one line of code is needed.
```

Question 27.

Once fully and correctly implemented what is the best case relative time complexity (Big-O value) of the method `sort`?

A) O(1)

B) O(n)

C) $O(n^2)$

D) O(n log n)

E) O(log n)

```
//Use the code for method sort to answer
//questions 24, 25, 26 and 27.

public static void sort(int list[],int left,
                        int right)
{
 if(left>=right) return;
 int x=left;
 int y=right;
 int z=<code 1>;
 while(x<y) {
   while(list[x]<z) x++;//line #1
   while(list[y]>z) y--;
   if(x<=y) {
     int t=list[x];
     list[x]=list[y];
     list[y]=t;
     x++;
     y--;
     }
   }
 <code 2>
 <code 3>
}
```

Question 28.

Which of the following must replace **<code>** in the code fragment shown on the right to ensure that the result of the expression will be a random whole number between 20 and 29 inclusive?

A) `*10+20`

B) `+10*20`

C) `*20+10`

D) `*10+29`

E) `*29+10`

```
(int)(Math.random()<code>)
```

Question 29.

Which of the following can replace **<code>** in the segment shown on the right?

A) `<String>`

B) `<Object>`

C) Will compile and run without replacing **<code>**.

D) A and B

E) A, B, and C

Question 30.

What is the output at **line #1** in the code segment shown on the right?

A) `[]`

B) `[turtle, cat, zebra]`

C) `[bird, antelope]`

D) `[bird, turtle, antelope, cat, zebra]`

E) `[dog, cat, zebra, turtle, monkey]`

Question 31.

What does **line #2** in the code segment shown on the right print?

A) `[dog, cat, zebra, turtle, monkey]`

B) `[dog, monkey]`

C) `[cat, zebra, turtle]`

D) `[cat, dog, monkey, turtle, zebra]`

E) `[]`

```
//Use the following code segment to answer
//questions 29, 30 and 31.

Queue<code> p=new LinkedList<code>();
p.add("bird");p.add("turtle");p.add("antelope");
p.add("cat");p.add("zebra");
ArrayList<code> list=new ArrayList<code>();
list.add("dog");list.add("cat");list.add("zebra");
list.add("turtle");list.add("monkey");
p.removeAll(list);
out.println(p);//line #1
out.println(list);//line #2
```

Question 32.

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

A) `5 8 5`

B) `5 8 8`

C) `4 8 7`

D) `4 7 7`

E) `5 8 11`

```
int c=5,d=7,e=3;
if(++d-c<=e||c--<d+c)
     e+=c;
else
     e+=d;
out.print(c+" "+d+" "+e);
```

Question 33.

What is printed by the code segment shown here?

```
String s1="supper";
String s2="superduper";
String s3=s1.substring(1, 4).equals(s2.substring(1, 4))?s1:s2;
out.print(s3);
```

A) uppupe

B) upp

C) upe

D) superduper

E) supper

Question 34.

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

A) [5, 8, 4, 1, 2, 3, 6, 7, 9]

B) [1, 2, 3, 4, 5, 6, 7, 8, 9]

C) [5, 8, 2, 4, 6, 7, 9, 1, 3]

D) [5, 8, 4, 1, 2, 6, 7, 9, 3]

E) [5, 8, 4, 2, 6, 7, 9, 1, 3]

```
int[] list= {5,8,4,6,2,9,7,1,3};
Arrays.sort(list, 3, 7);
out.print(Arrays.toString(list));
```

Question 35.

Which of the following best describes the value returned by `method`?

A) The number of vowels in the string `s` or -1 if `s` does not contain a vowel.

B) The index number of the first vowel contained in `s` or -1 if `s` does not contain a vowel.

C) The index value of each vowel in `s` or -1 if there are no vowels present.

D) The ordinal (1 based) position of the first vowel contained in `s` or -1 if `s` does not contain a vowel.

E) The number of times the string "aeiou" is found in `s` or -1 if it does not occur.

```
public static int method(String s) {
  s=s.toLowerCase();
  int x;
  for(x=0;x<s.length();x++)
    if(s.substring(x, x+1).matches("[aeiou]"))
      break;
  if(x!=s.length())
    return x+1;
  else
    return -1;
  }
```

Question 36.

Which of the following Boolean expressions is not equivalent to the others listed?

A) $A * (A + B)$

B) $A + A * B$

C) $A + 1$

D) $A * 1$

E) $A + 0$

Question 37.

The weight in a weighted graph is a property of the graph's _____?

A) edges

B) vertices

C) direction

D) cycle

E) spanning tree

Question 38.

Which of the following is the decimal equivalent of the signed 8-bit binary two's complement value shown on the right?

A) 121

B) -6

C) -134

D) 134

E) -122

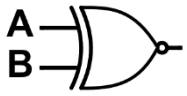
10000110

Question 39.

Rewrite this postfix expression using infix notation. A B C * + D –

Question 40.

Write the equivalent Boolean expression for the diagram shown here? You may use Java or generic notation to write the answer.



★ANSWER KEY – CONFIDENTIAL★

UIL COMPUTER SCIENCE – 2018 REGION

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

1) A	11) C	21) C	31) A
2) B	12) E	22) A	32) B
3) E	13) A	23) B	33) D
4) E	14) D	24) E	34) E
5) B	15) C	25) B	35) D
6) E	16) B	26) C	36) C
7) C	17) A	27) D	37) A
8) D	18) E	28) A	38) E
9) A	19) B	29) E	*39) A+B*C-D
10) B	20) D	30) C	*40) !(A^B) also $\overline{A \oplus B}$

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

Note: Correct responses are based on **Java SE Development Kit 8 (JDK 8)** from Sun Microsystems, 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 8 Standard Packages have been imported. Ignore any typographical errors and assume any undefined variables are defined as used.

Explanations:

1.	A	$AF_{16}=175_{10}$ $77_8=63_{10}$ 175+63=238_{10} Answer choices B, C, D, and E are all equal to 238_{10} $22201_3 = 235_{10}$ (1X1)+(2X9)+(2X27)+(2X81)=235
2.	B	-8%3+15%-4 = -2+3 = 1
3.	E	The escape sequences \" surround the result in quotes. The flag , (comma) makes the number have a comma separator. The flag ((parenthesis) encloses the number in parenthesis if it is negative. 7.2 rounds the number to two decimal places and places it into 7 spaces. When not enough space is allocated, printf will take the space that is needed.
4.	E	The substring method returns m. The two argument indexOf method will return the index of the first argument when it is found after the index number provided by the second argument. In this case, m does not appear in the string after index 20.
5.	B	F||F&&F^F= F||F&&F= F||F= F
6.	E	Math.round(x) returns long. long is promoted to float and double when needed.
7.	C	ASCII value of 'c' is 99. 99+12.75=111.75. Casting to char truncates to 111. ASCII number 111 = 'o'.
8.	D	n n%10 s 1593535253 3 1 159353525 5 6 15935352 2 11 1593535 5 16 159353 3 17 15935 5 22 1593 3 23 159 9 28 15 5 33 1 1 38
9.	A	k is incremented before the comparison to 5 is made, therefore, 0 is not printed.
10.	B	After the line b=a; both a and b point to the same array, array a. From this point any action taken on either array is taken on array a.
11.	C	next() returns a string.
12.	E	A produces an out of bounds error. B finds the sum of the evens. C produces an out of bounds error. D finds the sum of all the numbers except for 1.
13.	A	5|2*3=5|6. 101 OR 110 = 111 $111_2 = 7$
14.	D	32767-(-128)=32895
15.	C	5 5 3 5 3 8 5 6 3 8 5 6 4 8 5 6 4
16.	B	When the base case is reached the call stack is: l il uil gouil Pop each off the stack to get: liluilgouil
17.	A	For each row, if the current value in the row is less than the previous one, replace the current one with the previous one +1.

18.	E	Auto boxing automatically converts a primitive data type value to an object.
19.	B	Classes implement interfaces.
20.	D	When a class implements an interface, that class must implement all of the methods defined in the interface.
21.	C	An abstract method has a signature but is not implemented.
22.	A	i is a Cls object, c is a Cls object that has inherited from Intf.
23.	B	$7 = 00111_2$ Shift the bits to the left two spaces. That leaves 11100_2.
24.	E	The partition being sorted is bounded by the index values stored in left and right.
25.	B	This value will be swapped with the value found by the next line which will be smaller than the pivot.
26.	C	Divide this partition in two and recursively call the sort method on each of the new partitions. Since x and y have crossed paths in the code, ending the while loop, y is now the upperbound of the left partition and x is the lowerbound of the right partition.
27.	D	Best O(n log n), Average O(n log n), Worst $O(n^2)$
28.	A	Math.random returns 0.0 (inclusive) to 1.0 (exclusive). 0.0*10+20=0+20=20 0.99*10+20=9.9+20=29.9 which casts to 29
29.	E	Object is the cosmic super class. The elements in each collection are Strings. While it is greatly preferred that they are, collections do not strictly have to be parameterized.
30.	C	removeAll method removes everything in list from p.
31.	A	list is unaffected byt the call to removeAll.
32.	B	The left hand operand of the || operator evaluates to true. Because of short circuit evaluation the right hand side is never evaluated. Therefore c is not decremented.
33.	D	s1.substring(1,4) is “upp”. s2.substring(1,4) is “upe”. Ternary operator works like this: condition?true case:false case. “upp” is not equal to “upe” so false case is s2.
34.	E	This version of sort sorts the array beginning at the first argument and ending at one less than the second argument.
35.	D	[aeoiu] will match any of the vowels in the set. The loop breaks when a vowel is found. x stores the index number of that vowel. x+1 is the actual position of the vowel within the word when starting the count at one.
36.	C	A*(A+B)=A - Law of Absorption A+A*B=A - Law of Absorption A+1=1 – Law of Union A*1=A – Identity Law for AND A+0=A – Identity Law for OR
37.	A	The weight of an edge is the cost to travel along that edge from one vertex to another. The cost can be measured in money, time, distance or any other appropriate unit.
38.	E	Find the complement: 01111001. Add one to get 01111010. Convert to a decimal to get 122. Has to be negative because the sign bit was 1 in the original value. So, 10000110_2 = -122.
39.		A+B*C-D or B*C+A-D or A-D+B*C
40.		This is the logic gate for XNOR. Also accept $\overline{A \oplus B}$. Do not accept XNOR as an answer.