UIL Computer Science Written Test

2024 Invitational B

February/March 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 Invitational B

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 14_{16} and 101_{2}?

A) 1414_{16} **B)** 196_{10} **C)** 420_{5} **D)** 144_{8} **E)** 86_{12}

Question 2

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

A) `4` **B)** `5` **C)** `6` **D)** `7` **E)** `8`

```
out.print((25 + 7)/(12 % 7));
```

Question 3

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

A)
```
30AB
AB
ABAB
```

B)
```
30
3030
3030
```

C)
```
30
AB131
A1614B
```

D)
```
30
ABAB
A1614B
```

E)
```
30
21A16
A16
```

```
int A = 14;
int B = 16;
out.println(A + B);
out.print("A" + "B");
out.println('A' + 'B');
out.print("A" + B);
out.println(A + "B");
```

Question 4

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

A) `O` **B)** `ON` **C)** `T` **D)** `TON` **E)** `AN`

```
String St1 = "MICHIGAN";
String St2 = "WASHINGTON";
out.print(St2.substring(St1.length()));
```

Question 5

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

A) `true`

B) `false`

```
boolean A = true;
boolean B = false;
boolean C = A && B;
out.print(C || B || A);
```

Question 6

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

A) `-89.1` **B)** `-90.0` **C)** `89.1` **D)** `90` **E)** `98`

```
double T = Math.ceil(99.1);
double V = Math.sqrt(T);
out.print(V - T);
```

Question 7

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

A) `6` **B)** `6.0` **C)** `4` **D)** `4.0` **E)** `3`

```
double L = 29 / 10;
double M = L * 2.0;
double P = M + 1 / 2;
double Q = P - 0.2;
int A = (int) Q;
out.print(A);
```

Question 8

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

A) `8`

B) `11`

C) `12`

D) `17`

E) `23`

```
int B = 11;
if(B > 10)
  B += 5;
else
  B *= 2;
if(B < 14)
  B /= 2;
else
  B++;
out.print(B);
```

Question 9

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

A) `23 20 17 14 11 8 5`

B) `23 20 17 14 11 8 5 2`

C) `23 20 17 14 11 8`

D) `20 17 14 11 8 5`

E) `23 20 17 14 11 8 5 2`

```
for(int x = 23; x>=5; x=x-3)
  out.print(x + " ");
```

Question 10

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

A) `11` **B)** `14` **C)** `28` **D)** `31` **E)** `42`

```
int[] goat = new int[5];
goat[0] = 11;
goat[1] = 3;
for(int x=1; x<=4; x++)
  goat[x] = goat[x] + goat[x-1];
out.print(goat[4]);
```

Question 11

What is output by the code segment to the right?

A) `7`

B) `8`

C) `9`

D) `12`

E) `57`

```
String St = "12 0 5 3 2 8 7 6 9 4 0 1";
Scanner Sc = new Scanner(St);
int T = 0;
for (int x=1; x<=5; x++)
  {
    Sc.next();
    T = Math.max(T, Sc.nextInt());
  }
out.print(T);
```

Question 12

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

A) 10 13 16 19

B) 10 13 16 19 22 19 16 13 10

C) 10 13 16 19 16 13 10

D) 10 13 16 19 20 17 14 11

E) 10 13 16 19 22 20 17 14 11 8

```
for(int i=10; i<=20; i=i+3)
  out.print(i + " ");
for(int i=20; i>=10; i=i-3)
  out.print(i + " ");
```

Question 13

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

A) 10 **B)** 20 **C)** 40 **D)** 80 **E)** 160

```
int H = 10 << 2;
int J = H >> 4;
int K = H >> J;
out.print(K);
```

Question 14

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

A) 1 **B)** 2 **C)** 3 **D)** 7 **E)** 8

```
out.println(3 * 3 & 10 - 3);
```

Question 15

What is output by the code segment to the right?

A) 2

B) 3

C) 6

D) 12

E) 20

```
ArrayList<Integer> Stuff;
Stuff = new ArrayList<Integer>();
int[] List = {1,2,3,4,5,6,7};
for(int x=1; x<List.length; x++)
  Stuff.add(List[x-1]*List[x]);
out.print(Stuff.get(3));
```

Question 16

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

A) 22

B) 25

C) 29

D) 33

E) 43

```
String one = "5 1 2 9 2 6 7 4 1 7";
String two = "8 0 6 3 5 2 4 3 6 3";
String ten = "7 1 3 6 5 3 1 4 6 4";
Scanner A = new Scanner(one);
Scanner B = new Scanner(two);
Scanner C = new Scanner(ten);
int M = 0;
for(int x=1; x<=3; x++)
  {
    M += A.nextInt();
    B.next();
    M += B.nextInt();
    C.next(); C.next();
    M += C.nextInt();
  }
out.print(M);
```

Question 17

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

A) 33 **B)** 36 **C)** 39 **D)** 42 **E)** 45

```
int T = 42;
while(T>=36)
  T = T - 3;
out.print(T);
```

Question 18

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

A) 0 **B)** 6 **C)** 12 **D)** 24 **E)** 192

```
out.print(3 << 2 & 48 >> 2);
```

Question 19

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

A) 1 14 13 28 30 38 34 48 42 66 72 20

B) 1 14 13 30 22 24 36 34 50 52 74 22

C) 1 14 13 30 22 32 28 42 42 60 66 14

D) 1 14 13 30 22 32 28 42 46 64 70 20

E) 1 14 13 30 22 32 28 42 42 60 70 20

```
int[]red  = {1,3,5,7,9,0,2,4,6,8,10,14};
int[]blue = {9,8,7,6,5,4,3,2,1,7,12,15};
for(int x=1; x<=10; x++)
{
  red[x]  = blue[x-1];
  blue[x]  = red[x+1];
  red[x]  += blue[x];
  blue[x]  += red[x-1];
 }
for(int cello: red)
  out.print(cello + " ");
```

Question 20

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

A) 33 **B)** 55 **C)** 66 **D)** 67 **E)** 77

Question 21

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

A) 20 **B)** 21 **C)** 22 **D)** 28 **E)** 40

Question 22

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

A) 29 **B)** 45 **C)** 50 **D)** 57 **E)** 63

```
public static int shoe(int A)
{
  if (A > 10)
    return sock(A -3);
  if (A > 5)
    return shoe(A - 2) + A;
  return A*3;
 }

public static int sock(int B)
{
  if (B % 2 ==0)
    return sock(B-3)+ B;
  return B*5;
 }

// Client Code
out.print(sock(11)); // line #1
out.print(shoe(7)); // line #2
out.print(shoe(15)); // line #3
```

Question 23

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

A) 22

B) 23

C) 24

D) 25

E) 26

```
String Q = "ABCDEFGHIJKLM";
String R = "NOPQRSTUVWXYZ";
String T = R + Q;
for(int x=0; x<T.length(); x++)
{
 String Z = T.substring(x,x+1);
 if (Z.matches("[TEXAS]"))
   T=T.substring(0,x)+T.substring(x+1);
}
 out.print(T.length());
```

Question 24

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

A) 1 **B)** 2 **C)** 4 **D)** 6 **E)** 8

```
int T = 200;
int x = 1;
do
{
  T = T / x;
  x++;
 }
while (T>10);
out.print(T);
```

Question 25

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

A) 41 **B)** 44 **C)** 45 **D)** 54 **E)** 55

Question 26

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

A) 41 **B)** 44 **C)** 45 **D)** 54 **E)** 55

Question 27

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

A) 41 **B)** 44 **C)** 45 **D)** 54 **E)** 55

```
public static int[] Uno(int[]List)
 {
   int N = List.length;
   int[]NewList = new int[N-2];
   for (int x=1; x<List.length-1; x++)
     NewList[x-1] = List[x];
   return NewList;
 }

public static int[] Dos(int[]List)
 {
   int N = List.length;
   int[]NewList = new int[N-1];
   Arrays.sort(List);
   for (int x=1; x<List.length; x++)
     NewList[x-1] = List[x];
   return NewList;
 }

public static int Tres(int[]List)
 {
   int T = 0;
   for(int Bob:List)
     T += Bob;
   return T;
 }


 // Client Code
 int[]Roy = {9,2,8,4,10,7,6,1,3,5};
 out.print(Tres(Roy)); // Line #1
 out.print(Tres(Uno(Roy))); // Line #2
 out.print(Tres(Dos(Roy))); // Line #3
```

Question 28

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

A) 0 **B)** 6 **C)** 7 **D)** 8 **E)** 9

Question 29

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

A) 0

B) 3

C) 6

D) 7

E) 9

Question 30

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

A) 0

B) 3

C) 5

D) 7

E) 8

```
int[]List = {8,6,7,5,3,0,9};
PriorityQueue<Integer> A;
A = new PriorityQueue<Integer>();
Stack<Integer> B;
B = new Stack<Integer>();
ArrayList<Integer> C;
C = new ArrayList<Integer>();
for(int T:List)
{
  A.add(T); B.push(T); C.add(T);
}
A.remove(); B.pop(); C.remove(0);
A.remove(); B.pop(); C.remove(0);
A.remove(); B.pop(); C.remove(0);
out.print(A.remove()); // Line #1
A.remove(); B.pop(); C.remove(0);
out.print(B.pop());  // Line #2
A.remove(); B.pop(); C.remove(0);
out.print(C.get(0)); // Line #3
```

Question 31

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

A) 256 **B)** 512 **C)** 1024 **D)** 2048 **E)** 4096

```
int Num = 1;
for(int x = 1; x<=4; x++)
  for(int y = x; y<=4; y++)
     Num = Num<<1;
out.print(Num);
```

Question 32

How many levels?

A) 8 **B)** 10 **C)** 11 **D)** 12 **E)** 1999

In a Binary Search Tree consisting of 2000 nodes. What is the minimum number of levels the tree may have?

Example: An initially empty Binary Search Tree adding the elements B, then A, then C would have 2 levels.

Question 33

What of the following is **<u>not</u>** a possible output for the code to the right?

A) 40 **B)** 42 **C)** 48 **D)** 54 **E)** 56

```
int T = 0;
for (int x=1; x<=100; x=x*2)
  T += (int)(Math.random()*3 + 6);
out.print(T);
```

Question 34

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

A) `Bing 63`

B) `Burl 51`

C) `Rosemary 63`

D) `Bing 51`

E) `Nat 51`

Question 35

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

A) `Bing 63`

B) `Burl 51`

C) `Rosemary 63`

D) `Bing 51`

E) `Nat 51`

Question 36

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

A) `Bing 63`

B) `Burl 51`

C) `Rosemary 63`

D) `Bing 51`

E) `Nat 51`

```
public class Gold
 {
  public String Name = "Bing";
  public int Age = 63;

  public Gold(int A)
   {
     Name = "Burl";
     Age = A;
   }

  public Gold(String A)
   {
     Name = A;
   }

  public Gold(String A, int B)
   {
     Name = A;
     Age = B;
   }

  public Gold()
   {
   }

}

 //Client code
 Gold A = new Gold();
 Gold B = new Gold("Rosemary");
 Gold C = new Gold("Nat",51);
 // Line #1 Below
 out.print(A.Name + " " + A.Age);
 // Line #2 Below
 out.print(B.Name + " " + B.Age);
 // Line #3 Below
 out.print(C.Name + " " + C.Age);
```

Question 37

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

A) `112`

B) `122`

C) `172`

D) `721`

E) `741`

```
String St = "1";
St += "72";
int B = Integer.parseInt(St,8);
out.print(B);
```

Question 38

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

A) UNIVERSI

B) UNVERS

C) XTFOY

D) UNXVTRST

E) UNVRS

```
String St = "UNIVERSITYOFTEXAS";
String Answer = "";
int x, y;
for(x=0,y=16;y>=9;x++,y--)
  if (St.charAt(x)>St.charAt(y))
    Answer+=St.charAt(x);
out.print(Answer);
```

Question 39

What is the output of the code to the right. It is an integer. Write you answer in the blank for #39.

```
TreeSet<Integer> Cat;
Cat = new TreeSet<Integer>();
for(int x=1; x<=99; x=2*x+1)
  Cat.add(x%10);
out.print(Cat.size());
```

Question 40

How many different combinations will give T a value of true. One of them is (A=true, B=false, C=false, D=false) . Count all of the combinations that produce a value of true and write it in the blank for #40. Your answer will be an integer from 1-16.

```
boolean A,B,C,D,T;
// A, B, C, and D are assigned some values.
T = A && !B || C && !D;
```

★ANSWER KEY 2024 IB – CONFIDENTIAL★

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

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

** 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	You might recognize that 14_{16} and 101_2 convert to the base 10 numbers of 20 and 5. That product would be 100_{10}. 1414_{16} - not even close 196_{10} - nope 420_5 - 4*25 + 2*5 + 0*1 = 110 pretty close 144_8 - 1*64 + 4*8 + 4*1 = 100 We have a winner!!! 86_{12} - 8*12 + 6*1 = 102 The runner-up

2.	C	`(25 + 7)/(12 % 7)`
		`   32   /   5    =  6`

3.	C	`int A = 14;`
		`int B = 16;`
		`out.println(A + B);` Prints 30 then carriage return (next line)
		`out.print("A" + "B");` Prints AB but no carriage return
		`out.println('A' + 'B');` Prints 131 (65+66) then carriage return.
		`out.print("A" + B);` Prints A16 but no carriage return
		`out.println(A + "B");` Prints 14B

4.	B	`String St1 = "MICHIGAN";`
		`String St2 = "WASHINGTON";`
		`out.print(St2.substring(St1.length()));`
		The length of MICHIGAN is 8
		WASHINGTON.substring(8) is everything from position 8 until the end: "ON"

5.	A	`C || B || A`
		`false || false || true`
		One true is sufficient for a bunch of expressions separated by or's.

6.	B	`double T = Math.ceil(99.1);` T = 100.0 `double V = Math.sqrt(T);` V = 10.0 `out.print(V - T);` V - T = -90.0
7.	E	`double L = 29 / 10;` L = 2.0 `double M = L * 2.0;` M = 4.0 `double P = M + 1 / 2;` P = 4.0 (1 / 2 = 0) `double Q = P - 0.2;` Q = 3.8 `int A = (int) Q;` A = 3 `out.print(A);`

8.	D	`int B = 11;` B = 11
		`if(B > 10)`
		`  B += 5;` B = 16
		`else`
		`  B *= 2;`
		`if(B < 14)`
		`  B /= 2;`
		`else`
		`  B++;` B = 17
		`out.print(B);`

9.	A	`for(int x = 23; x>=5; x=x-3)`
		`  out.print(x + " ");`
		x starts at 23, keeps subtracting 3, and prints while x is greater than or equal to 5
		It prints 23 20 17 14 11 8 5

10.	B	`int[] goat = new int[5];` goat = {0,0,0,0,0}
		`goat[0] = 11;` goat = {11,0,0,0,0}
		`goat[1] = 3;` goat = {11,3,0,0,0}
		`for(int x=1; x<=4; x++)`
		`  goat[x] = goat[x] + goat[x-1];`
		x=1 goat = {11,14,0,0,0]
		x=2 goat = {11,14,14,0,0]
		x=3 goat = {11,14,14,14,0]
		x=4 goat = {11,14,14,14,14]
		`out.print(goat[4]);` 14

11.	B	`String St = "12 0 5 3 2 8 7 6 9 4 0 1";`
		`Scanner Sc = new Scanner(St);`
		`int T = 0;`
		x=1 skip 12 T=Math.max(0,0) T=0
		x=2 skip 5 T=Math.max(0,3) T=3
		x=3 skip 2 T=Math.max(3,8) T=8
		x=4 skip 7 T=Math.max(8,6) T=8
		x=5 skip 9 T=Math.max(8,4) T=8

12.	D	First loop prints 10 13 16 19 Second loop prints 20 17 14 11
13.	A	`int H = 10 << 2;` H = 40 `int J = H >> 4;` J = 2 `int K = H >> J;` K = 40 >> 2 K=10

14.	A	`3 * 3 & 10 - 3`
		Multiply first: `9 & 10 - 3`
		Subtract next: `9 & 7`
		`1001 & 0111 = 0001 = 1`

15.	E	`int[] List = {1,2,3,4,5,6,7};`
		Stuff = []
		x=1 Stuff = [2]
		x=2 Stuff = [2,6]
		x=3 Stuff = [2,6,12]
		x=4 Stuff = [2,6,12,20]
		x=5 Stuff = [2,6,12,20,30]
		x=6 Stuff = [2,6,12,20,30,42]
		`out.print(Stuff.get(3));` Print 20

16.	B	String one = "5 1 2 9 2 6 7 4 1 7";
		String two = "8 0 6 3 5 2 4 3 6 3";
		String ten = "7 1 3 6 5 3 1 4 6 4";
		int M = 0;
		x=1 Add 5 Skip 8 Add 0 Skip 7 Skip 1 Add 3 M=8
		x=2 Add 1 Skip 6 Add 3 Skip 6 Skip 5 Add 3 M=15
		x=3 Add 2 Skip 5 Add 2 Skip 1 Skip 4 Add 6 M=25

17.	A	T's values are 42, 39, 36, 33 then it exits the loop.
18.	C	3 << 2 & 48 >> 2 First do << 12 & 48 >> 2 Next do >> 12 & 12 Finally do & 1100 & 1100 = 1100 = 12

19.	C	int[]red = {1,3,5,7,9,0,2,4,6,8,10,14};
		int[]blue = {9,8,7,6,5,4,3,2,1,7,12,15};
		x=1 red={1,14,5,7,9,0,2,4,6,8,10,14} blue = {9,6,7,6,5,4,3,2,1,7,12,15}
		x=2 red={1,14,13,7,9,0,2,4,6,8,10,14} blue = {9,6,21,6,5,4,3,2,1,7,12,15}
		x=3 red={1,14,13,30,9,0,2,4,6,8,10,14} blue = {9,6,21,22,5,4,3,2,1,7,12,15}
		x=4 red={1,14,13,30,22,0,2,4,6,8,10,14} blue = {9,6,21,22,30,4,3,2,1,7,12,15}
		Continue the process 7 more steps.

20.	B	public static int shoe(int A)
		{
		if (A > 10)
		return sock(A -3);
		if (A > 5)
		return shoe(A - 2) + A;
		return A*3;
		}
		
		public static int sock(int B)
		{
		if (B % 2 ==0)
		return sock(B-3)+ B;
		return B*5;
		}
		
		// Client Code
		sock(11) = 55 (since 11 is odd)

21.	C	shoe(7) = shoe(5) + 7 = 22!!! shoe(5) = 15
22.	D	shoe(15) = sock(12) = 57!!! sock(12) = sock(9) + 12 = 57 sock(9) = 45

23.	A	T = "NOPQRSTUVWXYZABCDEFGHIJKLM" Go through each character of T. If a letter of "TEXAS" is there, it is removed and the process continues. It would seem that 5 letters are removed, but when the S is removed, the T "moves back" and is never checked. So only 4 letters are zapped. The resulting length is 22.
24.	E	T=200 200/1 = 200 200/2 = 100 100/3 = 33 33/4 = 8 8 causes the exit from the loop.
25.	E	Tres takes a list and returns the sum of the elements. int[]Roy = {9,2,8,4,10,7,6,1,3,5}; Tres(Roy) is the sum of the elements of Roy = 55
26.	A	Uno takes a list and returns the same list removing the first and last items. Tres takes a list and returns the sum of the elements. int[]Roy = {9,2,8,4,10,7,6,1,3,5}; Tres(Uno(Roy)) is the sum of all elements except for the 9 and the 5 = 41
27.	D	Dos takes a list, sorts it, and returns a list removing the first (smallest) value. Tres takes a list and returns the sum of the elements. int[]Roy = {9,2,8,4,10,7,6,1,3,5}; Tres(Dos(Roy)) is the sum of the elements except for the 1 = 54
28.	B	{8,6,7,5,3,0,9}; These numbers are placed in the Priority Queue A. Three are removed before the fourth one is removed and printed. The three that were removed would have been the 3 smallest items. The fourth item was 6
29.	D	{8,6,7,5,3,0,9}; These numbers are pushed onto the Stack B. Four are popped before the fifth one is popped and printed. The four that were removed would have been the 4 topmost items which were the last four pushed onto the Stack.. The fifth item popped was 7
30.	A	{8,6,7,5,3,0,9}; These numbers are added to the back of ArrayList C. The front item is removed five times. The five that were removed would have been the first 5 items in the list.. The sixth item we "got" was 0

31.	C	The line `Num = Num<<1` is invoked 10 times. So we double Num ten times. The result is 2^{10} which is 1024
32.	C	In a binary tree: 1 level holds at most 1 item. 2 levels hold at most 3 items. 3 levels hold at most 7 items. 4 levels hold at most 15 items. In general, N levels hold at most 2^N-1 items. 11 levels hold at most 2048 items.

33.	A	I `int T = 0;`
		`for (int x=1; x<=100; x=x*2)`
		`  T += (int)(Math.random()*3 + 6);`
		`out.print(T);`
		T is a value in the range [6,8].
		The loop repeats 7 times. (1,2,4,8,16,32,64)
		Thus, the smallest possible sum would be 42. The largest would be 56.
		40 is not in that range.

34.	A	In this case, the `Gold A = new Gold()` call would invoke the constructor with zero parameters. The default values are used - Bing 63.
35.	C	In this case, the `Gold A = new Gold("Rosemary")` call would invoke the second constructor. The Rosemary value is used with the default age. - Rosemary 63
36.	E	In this case, the `Gold C = new Gold("Nat",51)` call would invoke the two-parameter constructor. Both values passed in would be used.. - Nat 51
37.	B	$172_8 = \_\_\_\_\__{10}$ The answer is 122.
38.	E	This compares x and y characters from St. If the x character value is greater than the y character value, the x character value is appended to Answer. U vs. S U is added "U" N vs. A N is added "UN" I vs. X V vs. E V is added "UNV" E vs. T R vs. F R is added "UNVR" S vs. O S is added "UNVRS" I vs. Y
39.	4	The ones digit of 1,3,7,15,31,63 are added to Cat. Since Cat is a TreeSet, it does not hold duplicates. [1,3,5,7] The size is 4.

40.	7	**All combinations of 10** or **10** **0000** **0001** **0010 good** **0011** **0100** **0101** **0110 good** **0111** **1000 good** **1001 good** **1010 good** **1011 good** **1100** **1101** **1110 good** **1111**