UIL Computer Science Written Test

2024 Regional

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 REGIONAL

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 111101_2 and $3A_{16}$

A) 6423_8 **B)** 6512_8 **C)** 6722_8 **D)** 7002_8 **E)** 6572_8

Question 2

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

A) 5 **B)** 15 **C)** 25 **D)** 35 **E)** 45

```
out.print(23 + 34 % 7 * 2);
```

Question 3

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

A)
```
14
0
14
2
```

B)
```
2
0
2
2
```

C)
```
14
1
14
2
```

D)
```
14
0
-14
2
```

E)
```
14
0
-8
2
```

```
int A = 14;
int B = 16;
int C = 14;
int D = -22;

out.println(A % B);
out.println(A % C);
out.println(A % D);
out.println(B % C);
```

Question 4

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

A) M **B)** MA **C)** MAD **D)** MADN **E)** ADN

```
String St1 = "MARCH";
String St2 = "MADNESS";
int A = St1.indexOf("R");
int B = St2.indexOf("S");
out.print(St2.substring(A-2,B-2));
```

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

Question 6

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

A) 32 **B)** 48 **C)** 55 **D)** 56 **E)** 62

```
int Num = (2000>>8)<<(25>>3);
out.print(Num);
```

Question 7

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

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

```
double A = Math.sqrt(250);
double B = Math.cbrt(A);
double C = Math.ceil(B);
out.print((int)C);
```

Question 8

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

A) 20

B) 24

C) 25

D) 30

E) 32

```
int A = 5; int B = 3; int C = 7;
if(A>C)
  if(B>C)
    if(B>A)
      B=A+C;
    else
      C=A+B;
   else
     A=B+C;
 else
   A=A*3;
out.print(A+B+C);
```

Question 9

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

A) 3

B) 4

C) 5

D) 6

E) 8

```
int[] Numbers = {4,2,1,6,3,5,2,8};
int C = Numbers.length;
C+=Numbers[Numbers[4]];
C+=Numbers[Numbers[6]];
C/=2;
out.print(Numbers[C]);
```

Question 10

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

A) 9 **B)** 11 **C)** 15 **D)** 16 **E)** 21

```
int[] star = new int[20];
for(int x=1; x<10; x++)
 {
  star[x*2] = x*x;
  star[x*2-1]=star[x*2]-5;
 }
out.print(star[7]);
```

Question 11

What is output by the code segment to the right?

A) F

B) L

C) M

D) N

E) P

```
ArrayList<Character>France;
France = new ArrayList<Character>();
France.add((char)70);
for(int x=1; x<5; x++)
   France.add((char)(France.get(x-1)+2));
out.print(France.get(France.size()-1));
```

Question 12

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

A) -5 **B)** 7 **C)** 12 **D)** 15 **E)** 17

```
int A=5; int B=12; int C=7;
int D = (C>B) ? A+B : C-B;
out.print(D);
```

Question 13

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

A) 0 **B)** 4 **C)** 5 **D)** 14 **E)** 24

```
int N = 10;
int M = ++N << 2 ^ N++ >> 1 ;
out.print(M % N);
```

Question 14

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

A) -1 **B)** 0 **C)** 128 **D)** 255 **E)** 256

```
int A = Byte.MAX_VALUE;
int B = Byte.MIN_VALUE;
out.print(A - B);
```

Question 15

What is output by the code segment to the right?

A) 37

B) 48

C) 52

D) 54

E) 219

```
String Pan;
Pan = "8 C 6 A 7 B";
Scanner Sc = new Scanner(Pan);
int T = Sc.nextInt();
while(Sc.hasNext())
{
  String M = Sc.next();
  T += Integer.parseInt(M,16);
}
out.print(T);
```

Question 16

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

A) 12 **B)** 14 **C)** 16 **D)** 22 **E)** 43

```
int[]A = {8,6,7,5,3,0,9};
int[]B = {9,2,6,7,4,1,7};
int C = 0;
for(int x=0; x<A.length; x++)
  if(A[x]%2 == B[x]%2)
    C+= Math.max(A[x],B[x]);
out.print(C);
```

Question 17

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

A) 110 **B)** 192 **C)** 204 **D)** 214 **E)** 220

```
int[] Car = {22,66,37,34,5,29,43,21};
for(int x=0; x<Car.length-1; x=x+1)
 if(Car[x]>Car[x+1])
   Car[x+1] = Car[x+1]/2;
int G = 0;
for(int x=0; x<Car.length-1; x=x+1)
  G += Car[x];
out.print(G);
```

Question 18

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

A) IJK **B)** JKL **C)** KLM **D)** LMN **E)** MNO

```
String St = "ABCDEFGHIJKLMNOPQRST";
String A1 = St.substring(4);
String A2 = A1.substring(5,10);
int L = A2.length();
String A3 = A2.substring(1,L-1);
out.print(A3);
```

Question 19

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

A) 16 **B)** 18 **C)** 20 **D)** 22 **E)** 24

```
HashSet<Integer>House;
House = new HashSet<Integer>();
for(int x=20; x<100; x=x+1)
  House.add(x / 5);
out.print(House.size());
```

Question 20

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

A) 32 **B)** 33 **C)** 39 **D)** 40 **E)** 41

Question 21

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

A) 72 **B)** 75 **C)** 78 **D)** 81 **E)** 96

Question 22

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

A) 245 **B)** 251 **C)** 252 **D)** 253 **E)** 256

```
public static int Fun(int A)
 {
  if (A > 100)
    return Fun(A - A%2 - 20) + A/2;
  if (A > 50)
    return Fun(A - A%3 - 10) + A/3;
  if (A > 25)
    return Fun(A - A%4 - 5) + A/4;
  return A;
 }

  // Client Code
  out.println(Fun(32));   // line #1
  out.println(Fun(64));   // line #2
  out.println(Fun(128));  // line #3
```

Question 23

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

A) [66,74,82,98]

B) [26,42,74,82,98]

C) [42,82,98]

D) [42,74,82,98]

E) [42,74,98]

```
ArrayList<Integer>Mel;
Mel = new ArrayList<Integer>();
for(int x=10; x<100; x=x+8)
  Mel.add(x);
for(int x=0; x<Mel.size(); x=x+1)
 {
  if(Mel.get(x)%10==0)
    Mel.remove(x);
  if(Mel.get(x)%11==0)
    Mel.remove(x);
 }
for(int x=0; x<Mel.size(); x=x+1)
 {
    int A = Mel.get(x)/10;
    int B = Mel.get(x)%10;
    if (A<B)
      Mel.remove(x);
 }
 out.print(Mel);
```

Question 24

Which statement to the right would generate a random integer from 25 to 35 inclusive? [25,35].

A) A **B)** B **C)** C **D)** D **E)** E

```
int A = (int)(Math.random()*25) + 35;
int B = (int)(Math.random()*10) + 25;
int C = (int)(Math.random()*35) + 25;
int D = (int)(Math.random()*11) + 25;
int E = (int)(Math.random()*10) + 35;
```

Question 25

In the class to the right, what word should replace the **<1.???>** on line #1?

A) int
B) private
C) public
D) Regional
E) return
F) String
G) super
H) void

Question 26

In the class to the right, what word should replace the **<2.???>** on line #2?

A) int
B) private
C) public
D) Regional
E) return
F) String
G) super
H) void

Question 27

In the class to the right, what word should replace the **<3.???>** on line #3?

A) int
B) private
C) public
D) Regional
E) return
F) String
G) super
H) void

```
public class Regional
  {
   private String A;
   private String B;
   private int C;

   public <1.???>()          //Line #1

     {
     A = "Austin";
     B = "Texas";
     C = 2024;
     }
   public String getA()
     {
     return A;
     }
   public String getB()
     {
     return B;
     }
   public <2.???> getAll()   //Line #2
     {
     return A + " " + B + " in" + C;
     }
   public <3.???> Event()    //Line #3
     {
     System.out.print("Computer Science");
     }
}
```

Question 28

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

A) 11 **B)** 6 **C)** 7 **D)** 8 **E)** 9

Question 29

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

A) [27, 28, 51, 43, 63, 66, 73, 63, 50]

B) [27, 28, 43, 51, 63, 66, 73, 63, 50]

C) [27, 43, 28, 50, 63, 51, 73, 63, 66]

D) [27, 28, 51, 43, 63, 63, 73, 66, 50]

E) [27, 28, 43, 50, 51, 63, 63, 66, 73]

Question 30

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

A) [27, 28, 51, 43, 63, 66, 73, 63, 50]

B) [27, 28, 43, 51, 63, 66, 73, 63, 50]

C) [27, 43, 28, 50, 63, 51, 73, 63, 66]

D) [27, 28, 51, 43, 63, 63, 73, 66, 50]

E) [27, 28, 43, 50, 51, 63, 63, 66, 73]

```
PriorityQueue<Integer> W;
W = new PriorityQueue<Integer>();
W.add(27); W.add(11); W.add(63);
W.add(19); W.add(25); W.add(51);
W.add(73); W.add(50); W.add(43);
W.add(63); W.add(66); W.add(28);
out.println(W.peek());   //Line #1

W.remove(); W.remove(); W.remove();
out.println(W);          //Line #2
W.add(W.remove());
out.println(W);          //Line #3
```

Question 31

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

A) 210 **B)** 280 **C)** 360 **D)** 450 **E)** 550

```
int C = 0;
for(int x = 0; x<100; x++)
  C += x % 10;
out.print(C);
```

Question 32

Find the value.

A) 15 **B)** 31 **C)** 63 **D)** 64 **E)** 127

$2^{16} = 65536$

What is the value of….

$65535 >> 10$

Question 33

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

A) 2 **B)** 3 **C)** 4 **D)** 6 **E)** 8

```
int[]V = {3,4,7,9,12,34,56,88,91};
int A = Arrays.binarySearch(V,88);
int B = Arrays.binarySearch(V,10);
out.print(A+B);
```

Question 34

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

A) CAT

B) OWL

C) FISH

D) RHINO

E) EEL

Question 35

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

A) EEL

B) FROG

C) RHINO

D) FISH

E) HIPPO

Question 36

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

A) 6

B) 12

C) 14

D) 17

E) 18

```
TreeMap<Integer,String> UIL;
UIL = new TreeMap<Integer,String>();
UIL.put(20, "DOG");
UIL.put(35, "CAT");
UIL.put(12, "OWL");
UIL.put(18, "EEL");
for(int x=1; x<=5; x++)
  UIL.put(x, "FROG");
for(int x=3; x<=8; x++)
  UIL.put(x, "HIPPO");
UIL.put(12, "RHINO");
UIL.put(35, "FISH");

out.print(UIL.get(12));  //Line #1
out.print(UIL.get(5));   //Line #2
out.print(UIL.size());   //Line #3
```

Question 37

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

A) [12, 15, 20]

B) [15, 20, 12]

C) [20, 12, 15]

D) [20, 15, 12]

E) [15, 12, 20]

```
ArrayList<Integer>K;
K = new ArrayList<Integer>();
K.add(20); K.add(12); K.add(15);
for(int i=1; i<=11; i++)
  for(int x=0; x<K.size()-1; x++)
  {
    int T = K.get(x);
    K.set(x,K.get(x+1));
    K.set(x+1, T);
  }
out.print(K);
```

Question 38

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

A) 21

B) 25

C) 29

D) 33

E) 35

```
String One = "UIL";
String Two = "FUN";
int C = 0;
for(int x=0; x<3; x++)
 {
   String A = One.substring(x,x+1);
   String B = Two.substring(x,x+1);
   int D = A.compareTo(B);
   C += Math.abs(D);
 }
out.print(C);
```

Question 39

What is the sum of all the numbers that remain on the stack to the right at the end of the last statement. Write your answer in blank #39 on the answer sheet.

```
The stack is initially empty.
Push 9
Push 1
Push 8
Push 2
Pop
Pop
Push 7
Push 3
Push 6
Push 4
Peek
Peek
Pop
Push 5
Push 10
Pop
Process Complete
```

Question 40

Evaluate the prefix expression to the right.

```
+ 7 * 3 + - * / 8 2 6 5 1
```

Write the number in blank #40.

★ANSWER KEY – CONFIDENTIAL★

UIL COMPUTER SCIENCE – 2024 Region

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

1) **C**	11) **D**	21) **D**	31) **D**
2) **D**	12) **A**	22) **D**	32) **C**
3) **A**	13) **C**	23) **B**	33) **A**
4) **C**	14) **D**	24) **D**	34) **D**
5) **A**	15) **D**	25) **D**	35) **E**
6) **D**	16) **D**	26) **F**	36) **B**
7) **B**	17) **D**	27) **H**	37) **B**
8) **C**	18) **C**	28) **A**	38) **C**
9) **E**	19) **A**	29) **C**	*39) **31**
10) **B**	20) **B**	30) **A**	*40) **67**

** 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.	C	Change everything to base 8 first. 111101_2 becomes 75_8 and $3A_{16}$ becomes 111010_2 which is 72_8 Multiply 75 and 72 using base 8 rules. 5 x 2 = 12 70 x 2 = 160 70 x 5 = 430 70 x 70 = 6100 6100 + 430 = 6530 6530 + 160 = 6710 6710 + 12 = 6722

2.	D	`23 + 34 % 7 * 2`
		`23 + 6 * 2`
		`23 + 12 = 35`

3.	A	`int A = 14; int B = 16; int C = 14;int D = -22;` `out.println(A % B);` Prints 14 `out.println(A % C);` Prints 0 `out.println(A % D);` Prints 14 `out.println(B % C);` Prints 2

4.	C	`String St1 = "MARCH";`
		`String St2 = "MADNESS";`
		`int A = St1.indexOf("R");` A=2
		`int B = St2.indexOf("S");` B=5
		`out.print(St2.substring(A-2,B-2));` St2.substring(0,3) = "MAD"

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

6.	D	`int Num = (2000>>8)<<(25>>3);`
		`7 << 3`
		`56`

7.	B	`double A = Math.floor(8.7);` A = 8.0 `double A = Math.sqrt(250);` A is more than 15, less than 16 `double B = Math.cbrt(A);` B is more than 3, less than 4 `double C = Math.ceil(B);` C=4.0 `out.print((int)C);` 4
8.	C	`int A = 5; int B = 3; int C = 7;` Only the last else is triggered since A is not greater than C. 15 + 3 + 7 = 25

9.	E	`int[] Numbers = {4,2,1,6,3,5,2,8};` `int C = Numbers.length;` C = 8 `C+=Numbers[Numbers[4]];` C = 8 + 6 = 14 `C+=Numbers[Numbers[6]];` C = 14 + 1 = 15 `C/=2;` C = 7 `out.print(Numbers[C]);` C[7] = 8

10.	B	`star[x*2] = x*x;`
		`star[x*2-1]=star[x*2]-5;`
		When x is 4, star[8] = 16
		star[7] becomes 11

11.	D	This creates the ArrayList ['F','H','J','L','N'] The last element is 'N'

12.	A	`int A=5; int B=12; int C=7;`
		`int D = (C>B) ? A+B : C-B;`
		Translated
		`int D;`
		`if(C>B)`
		`  D=A+B;`
		`else`
		`  D=C-B;`
		D is -5

13.	C	`int N = 10;`
		`int M = ++N << 2 ^ N++ >> 1 ;` N = 12
		`11 << 2 ^ 11 >> 1`
		`44 ^ 5`
		`101100 ^ 000101 = 101001` M = 41
		`out.print(M % N);`
		41 % 12 = 5

14.	D	`int A = Byte.MAX_VALUE;`
		`int B = Byte.MIN_VALUE;`
		`out.print(A - B);`
		127 - (-128) = 255

15.	D	`String Pan;`
		`Pan = "8 C 6 A 7 B";`
		`Scanner Sc = new Scanner(Pan);`
		`int T = Sc.nextInt();` T = 8
		`while(Sc.hasNext())`
		`{`
		`  String M = Sc.next();`
		`  T += Integer.parseInt(M,16);` T = 8 + 12 + 6 + 10 + 7 + 11
		`}`
		`out.print(T);` 54

16.	D	`int[]A = {8,6,7,5,3,0,9};`
		`int[]B = {9,2,6,7,4,1,7};`
		
		If the corresponding elements in the lists are either both odd or both even, the larger number is added to the sum.
		6 + 7 + 9 = 22

17.	D	Car becomes {22, 66, 18, 34, 2, 29, 43} The sum of the elements is 214

18.	C	`String St = "ABCDEFGHIJKLMNOPQRST";`
		`String A1 = St.substring(4);` A1 = "EFGHIJKLMNOPQRST"
		`String A2 = A1.substring(5,10);` A2 = "JKLMN"
		`int L = A2.length();`
		`String A3 = A2.substring(1,L-1);`
		`out.print(A3);` A3 is all but the first and last of A2
		KLM

19.	A	Only the numbers 4 through 19 are added to the set. A set allows no duplicates. The size is 16
20.	B	Fun(32) = Fun(27) + 8 = 33 Fun(27) = Fun(19) + 6 = 25 Fun(19) = 19
21.	D	Fun(64) = Fun(53) + 21 = 81 Fun(53) = Fun(41) + 17 = 60 Fun(41) = Fun(35) + 10 = 43 Fun(35) = Fun(27) + 8 = 33 Fun(27) = Fun(19) + 6 = 25 Fun(19) = 19
22.	D	Fun(128) = Fun(108) + 64 = 253 Fun(108) = Fun(88) + 54 = 189 Fun(88) = Fun(77) + 29 = 135 Fun(77) = Fun(65) + 25 = 106 Fun(65) = Fun(53) + 21 = 81 Fun(53) = Fun(41) + 17 = 60 Fun(41) = Fun(35) + 10 = 43 Fun(35) = Fun(27) + 8 = 33 Fun(27) = Fun(19) + 6 = 25 Fun(19) = 19
23.	B	List originally has [10,18,26,34,42,50,58,66,74,82,90,98] Zap all multiples of 10 and multiples of 11 [18,26,34,42,58,74,82,98] Now, zap all elements where 10s digit is less than 1s digit But while 18, 34, and 58 are removed… 26, and 42 survive since they move back when item before is removed. [26,42,74,82,98]

24.	D	int D = (int)(Math.random()*RANGE) + SMALLEST; D is the answer because the range is 11 and the smallest value is 25
25.	D	Line #1 - Constructors must be public Line #2 - The method is returning a String Line #3 - The method is a void method, returning no values
26.	F	
27.	H	
28.	A	The smallest item is "seen" with peek on a Priority Queue… 11
29.	C	Three items are removed…11, 19, and 25. With each, the Priority Queue changes.
30.	A	27 is removed, but when added back in, it makes it's way to the top. The list does not always end up as it was before the removal.
31.	D	The numbers 0+1+2+3+4+5+6+7+8+9 = 45. 45 is added to C ten times giving us 450
32.	C	2^{16} = 1000000000000000 (1 + 16 zeros) 2^{16} - 1 = 111111111111111 (16 ones) Right shift 10 gives us only 6 ones $111111_2 = 63_{10}$

33.	A	`int[]V = {3,4,7,9,12,34,56,88,91};`
		`int A = Arrays.binarySearch(V,88);`
		`int B = Arrays.binarySearch(V,10);`
		A = 7 the position of the found item.
		B = -5 since the item would have been between the 9 and 12
		7 + (-5) = 2

34.	D	The last item to be put in position 12 is RHINO
35.	E	While FROG was first placed in position 5, HIPPO took its place.
36.	B	The only keys are 20,35,12,18,1,2,3,4,5,6,7,8 = 12 keys
37.	B	`[20,12,15]` The loop performs two swaps moving item 0 to the end. Item 1 moves to position 0 Item 2 moves to position 1 For every 3 times the loop is repeated, the list goes back to its original order. There are 11 iterations, therefore the list should look as it would after 2 iterations. [15,20,12]

38.	C	`String One = "UIL";`
		`String Two = "FUN";`
		The process compares corresponding letters in the two strings, finding the difference in their ASCII values - or the difference in their positions in the alphabet. Those positive differences are added together.
		U,F is 15
		I,U is 12
		L,N is 2
		Sum is 29

39.	31	The resulting stack is [9, 1, 7, 3, 6, 5]
40.	67	+ 7 * 3 + - * / 8 2 6 5 1 + 7 * 3 + - * 4 6 5 1 + 7 * 3 + - 24 5 1 + 7 * 3 + 19 1 + 7 * 3 20 + 7 60 67