University Interscholastic League

COMPUTER SCIENCE WRITTEN TEST STUDY PACKET 2014-2015

A PACKAGE OF PROGRAMMING PRACTICE MATERIALS IS AVAILABLE FOR PURCHASE THROUGH THE UIL ONLINE STORE:

http://store.uiltexas.org

VISIT THE UIL WEBSITE FOR LINKS TO ADDITIONAL COMPUTER SCIENCE RESOURCES AND STUDY AND PRACTICE MATERIAL:

http://www.uiltexas.org/academics/computer-science

University Interscholastic League
Computer Science Competition

Number 143 (Invitational A - 2014)

General Directions:

1) **DO NOT OPEN EXAM UNTIL TOLD TO DO SO.**

2) **NO CALCULATOR OF ANY KIND MAY BE USED.**

3) There are 40 questions on this contest exam. You have 45 minutes to complete this contest. If you are in the process of actually writing an answer when the signal to stop is given, you may finish writing that answer.

4) Papers 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 paper until told to do otherwise. Use this time to check your answers.

5) All answers must be written on the answer sheet/Scantron card provided. Indicate your answers in the appropriate blanks provided on the answer sheet or on the Scantron card. Clean erasures are necessary for accurate Scantron grading.

6) You may place as many notations as you desire anywhere on the test paper, but not on the answer sheet or Scantron card, which are reserved for answers only.

7) You may use additional scratch paper provided by the contest director.

8) All questions have ONE and only ONE correct (BEST) answer. There is a penalty for all incorrect answers.

9) A reference to commonly used Java classes is provided at the end of the test, and you may use this reference sheet during the contest. You may detach the reference sheets from the test booklet, but DO NOT DO SO UNTIL THE CONTEST BEGINS.

Scoring:

1) All questions will receive 6 points if answered correctly; no points will be given or subtracted if unanswered; 2 points will be deducted for an incorrect answer.

Note: Correct responses are based on Java, **J2sdk v 1.7.25**, from Sun Microsystems, Inc. All provided code segments are intended to be syntactically correct, unless otherwise stated (i. e. `error` is an answer choice) and any necessary Java 2 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…** ***import static java.lang.System.*;***

QUESTION 1

Which of these is NOT equivalent to $11110_2 + 11011_2$?

A. 57_{10} B. 71_8 C. 39_{16} D. 111101_2 E. All are equivalent

QUESTION 2

What is output by the code to the right?

A. `19 32` B. `32 13`

C. `19 13` D. `32 19`

E. There is no output due to a compile error.

```
long b = 19;
int c = 13;
b+=c;
out.println(b+" "+c);
```

QUESTION 3

What is output by the code to the right?

A. `4` B. `3` C. `4.0`

D. There is no output due to a compile error.

E. There is no output due to a runtime error.

```
Integer [] list = {1,2,3,4.0};
out.println(list[3]);
```

QUESTION 4

What is output by the code to the right?

A. `4 3 2 1` B. `4 3 2`

C. `5 4 3 2` D. `5 4 3`

E. There is no output.

```
int j = 5;
do
{
  out.print(--j + " ");
}
while (j>1);
```

QUESTION 5

What is output by the code to the right?

A. `gBad` B. `reak` C. `Brea` D. `a` E. `k`

```
String s = "BreakingBad";
out.println(s.charAt(4));
```

QUESTION 6

What is output by the code to the right?

A. `aead` B. `abcd`

C. `abeb` D. `cecd`

E. There is no output.

```
char [] list1 = {'a','b','c','d'};
char [] list2 = list1;
list2[2] = 'e';
list1[3] = list2[1];
for(char a:list1)
     out.print(a);
```

QUESTION 7

What is output by the code to the right?

A. `false false` B. `false true`

C. `true false` D. `true true`

E. There is no output due to a runtime error.

```
boolean p = true;
boolean q = true;
p = p^q;
out.println(p + " " + q);
```

QUESTION 8

What is output by the code to the right?

A. `yum` B. `yumyom`

C. `burp` D. `chomp`

E. `yumyomchompburp`

```
String s1 = "sweet";
switch(s1)
{
  case "sweet":out.print("yum");
  case "sour" :out.print("yom");
               break;
  case "spicy":out.print("chomp");
  default     :out.print("burp");
}
```

QUESTION 9

What is output by the code to the right?

A. 3.1 B. 5.2 C. 8.3

D. 2.1 E. 2.5

```
out.println(Math.max(5.2,3.1));
```

QUESTION 10

What is output by the code to the right?

A. 999 B. 333

C. 342 D. 101010

E. 231

```
int[][]grid={{1,2,3},{4,5,6,7},
             {8,9}};
out.println(grid[0].length + "" +
            grid[1].length + "" +
            grid[2].length);
```

QUESTION 11

Which of the following correctly replaces **<statement1>** in the Guitar class definition on the right ?

A. public void

B. public int

C. private void

D. private int

E. public static int

QUESTION 12

Which of the following correctly replaces **<statement2>** in the Guitar class definition on the right ?

A. ( );

B. (int n)

C. ( )

D. (String s)

E. (int n);

QUESTION 13

Which of the following correctly replaces **<statement3>** in the Guitar class definition on the right ?

A. type = s;

B. numStrings = n;

C. return type;

D. return numStrings;

E. return 6;

```
class Guitar
{
  private String type;
  private int numStrings;
  public Guitar()
  {
     type = "acoustic";
     numStrings = 6;
  }
  public Guitar(int n)
  {
     this();
     numStrings = n;
  }
  public Guitar(int n, String s)
  {
     this(n);
     type = s;
  }
  public String toString()
  {
     return type + ": " +
        numStrings + " string";
  }
<statement1>getNumStrings<statement2>
  {
          <statement3>
  }
}

////////////////////////////////
////client code
Guitar g = new Guitar(5,"bass");
out.println(g);
```

QUESTION 14

What is output by the code to the right?

A. 7 B. 9

C. 15 D. 12

E. 31

```
int d = 25;
d = d | 15 & 7;
out.println(d);
```

QUESTION 15

What is output by the code to the right?

A. 99 B. 8 C. 7

D. 100 E. 0

```
int e = 0, f = 1;
while(f<100){
  e++;
  f*=2;
}
out.println(e);
```

QUESTION 16

Which term best describes the variable type for **a** in the client code shown?

A. actual parameter

B. formal parameter

C. instance field

D. class variable

E. temporary variable

QUESTION 17

What is output by the client code to the right?

A. 30 1 7

B. 30 1 1 7

C. 30 1 6 1 9 7

D. 30 1 6 45 1 9 35 1 7

E. There is no output due to a syntax error.

```
static void stuff(int x)
{
  if(x%2==0)
     out.print(x*5+" ");
  else
  if(x%3==0)
     out.print(x/5+" ");
  else
     out.print(x+" ");
}
//client code
int a = 6;
stuff(a);
a+=3;
stuff(a);
a-=2;
stuff(a);
```

QUESTION 18

Which of these statements will return the substring "Probe"?

I. `s.substring(7,12);`
II. `s.substring(8);`
III. `s.substring(8,13);`
IV. `s.substring(7,13);`
V. `s.substring(7);`

A. I and V only
B. II only
C. III only
D. II and III only
E. IV only

```
String s = "Cassini_Probe";
```

QUESTION 19

What is output by the code to the right?

A. 2.9 B. 4.5 C. 19.5 D. 4.9 E. 3.2

```
long k = 12;
int m = 5;
double p = 2.5;
out.println(p+k/m);
```

QUESTION 20

What is output by the code to the right?

A. 001 010 101 111

B. 000 010 100 111

C. 001 010 101 110

D. 001 011 100 110

E. 000 010 101 111

```
for(int p = 0; p <= 1; p++)
 for(int q = 0; q <= 1; q++)
  out.print(""+p+q+(p|q&p)+" ");
```

QUESTION 21

Based on the value of x in the code on the right, which of the following statements will output only the value 6 ?

I. `out.println(x%1000/100);`
II. `out.println(x/100%10);`
III. `out.println(x/1000%10);`

A. I only
B. II only
C. III only
D. I and II only
E. I and II and III

```
int x = 49627;
```

QUESTION 22

What is output by the code to the right?

A. `360.0` B. `180.0` C. `90.0`
D. `45.0` E. `0.0`

```
double d = Math.toDegrees(Math.PI*2);
out.printf("%.1f\n",d);
```

QUESTION 23

What is output by the code to the right?

A. `2147483647`
B. `-2147483648`
C. `11110000000000000000000000000000` (4 1s, 32 zeroes)
D. `11111111111111111111111111111111` (32 1s)
E. `1111`

```
int x = 15 << 32;
String s = Integer.toBinaryString(x);
out.println(s);
```

QUESTION 24

What is output by the code to the right?

A. `true0.0`
B. `true2.3`
C. `true3.1`
D. `false2.3`
E. `false4.2`

```
ArrayList <Double> list;
list = new ArrayList<Double>();
out.print(list.isEmpty());
list.add(2.3);
list.set(0,4.2);
list.add(3.1);
list.remove(0);
out.print(list.get(0));
```

QUESTION 25

Find f(12,6) according to the recursive function definition shown on the right. You may use the space below to do your work.

f(12,6) =

$$f(x,y) = \begin{cases} f(x-y,y-1)+2 & \text{when } x>y \\ x+y & \text{otherwise} \end{cases}$$

A. `5` B. `6` C. `7`
D. `9` E. `12`

QUESTION 26

What is output by the code to the right?

A. `Fry` B. `FryFa`

C. `FreettyFall` D. `FreeFallinTomPetty`

E. There is no output due to a compile error

```
String s = "FreeFallinTomPetty";
String [] ar = s.split("[elt]+");
out.println(ar[0]+ar[ar.length-1]
          +ar[1]);
```

QUESTION 27

What is output by the code to the right?

A. `1` B. `33`

C. `100` D. `bad`

E. `breaking`

```
String bb = (100%3==0)?"breaking"
                      :"bad";
out.println(bb);
```

QUESTION 28

What is output by the code to the right?

A. `false` B. `-8`

C. `8` D. `-1`

E. `1`

```
String s = "KarelJRobot";
String t = "Kilamanjaro";
out.println(s.compareTo(t));
```

QUESTION 29

A. `10` B. `20`

C. `ten` D. `sepuluh`

E. `tensepuluh`

```
Map<Integer,String> m =
  new HashMap<Integer,String>();
m.put(10,"ten");
m.put(14,"fourteen");
m.put(9,"nine");
m.put(10,"sepuluh");
out.println(m.get(10));
```

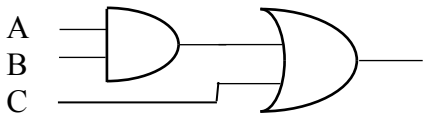
QUESTION 30

Which of the following logical statements is represented by the digital electronics diagram on the right ?

A. A && B || C B. A || B && C

C. A ^ B || C D. A || B ^ C

E. A && B ^ C



QUESTION 31

On the right is a boolean expression using generic notation. Which of the expressions below represents the simplest form of this expression ? (Note : * means AND, + means OR)

A. $\overline{A}$ B. 0 C. $\overline{A} * \overline{B}$ D. $\overline{A}(\overline{A}*\overline{B})$ E. $\overline{A}+\overline{B}$

$$\overline{A}\,\overline{(A + B)}$$

(this translates to ***"not A and not (A or B)"***)

QUESTION 32

In a typical binary search process, in how many steps will the value 5 be found in the array shown on the right?

A. 3 B. 4

C. 5 D. 6

E. 7

```
0 1 2 3 4 5 6 7 8 9 10 11 12 13
```

QUESTION 33

Which statement below best describes the minimum required <implementation> of class B for the class structure shown on the right?

A. class B is only required to define method **`one()`**.
B. class B is not required to implement anything.
C. class B is required to implement method **`one()`** and override method **`two()`**.
D. class B is only required to override method **`two()`**.
E. This class structure is invalid.

QUESTION 34

Suppose all is correctly defined with this class structure so that method one()returns the value 4. What is the output for the client code shown on the right?

A. `0` B. `5`

C. `20` D. `40`

E. There is no output due to a runtime error.

```
abstract class A
{
  int x = 2;
  abstract int one();
  int two()
  {
     return 5;
  }
}
class B extends A
{
  //<implementation>
}

//////////client code///////////////
B bop = new B();
out.println(bop.one()*bop.two()
              *bop.x);
```

QUESTION 35

Which of the following is an **INVALID** class B definition?

I.
```
class B extends A{
      int one(){
         return 4;
      }}
```
II.
```
class B extends A{
      x=1;
      int one(){
         return 4;
      }}
```
III.
```
class B extends A{
      int one(){
         return 4;
      }
      int two(){
         return 6;
      }}
```
IV.
```
class B extends A{
      int x = 4;
      int one(){
         return 4;
      }
      int two(){
         return 6;
      }}
```

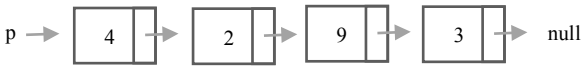
A. I is invalid
B. II is invalid
C. III is invalid
D. IV is invalid
E. All of these are valid

QUESTION 36

Suppose a linked list has been implemented as shown in the diagram on the right, with public fields **`data`** and **`next`**. What is the output of the statement below?

```
out.print(p.next.data);
```

A. 2 B. 3 C. 4 D. 5 E. 9



QUESTION 37

What is output by the code to the right?

A. 3null

B. 3false

C. 3true

D. 4false

E. 4true

```
Set<Integer> sa = new
  TreeSet<Integer>();
sa.add(4);
sa.add(5);
sa.add(4);
sa.add(6);
sa.add(7);
sa.remove(6);
out.print(sa.size());
out.println(sa.contains(6));
```

QUESTION 38

What is the output of this code if the value of ***<keyboard integer input>*** is 3.14?

A. Bad data.

B. All is good.

C. Bad data. All is good.

D. There is no output.

E. There is no output due to a runtime error.

```
int tx;
try{
  tx = <keyboard integer input>;
}
catch(Exception ee){
  out.print("Bad data. ");
}
finally{
  out.print("All is good. ");
}
```

QUESTION 39

On the right is a binary tree implementing a min heap, with the 1 in position 0, the 7 in position 1, and the 5 in position 2. The last element added was a 2. In what position does the value 2settle when the min heap is reestablished in the sifting up process?

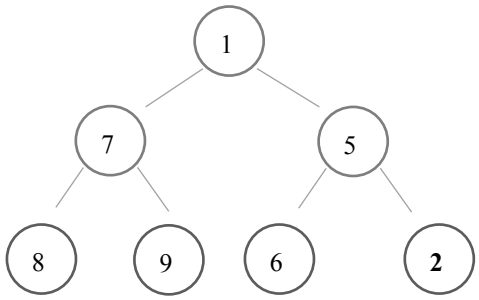
A. position 0

B. position 1

C. position 2

D. position 5

E. position 6

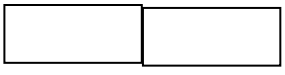


QUESTION 40

OPEN ENDED QUESTION – Using the ***enqueue*** and ***dequeue*** sequence given on the right, process the commands shown into a standard queue and indicate the ***last value dequeued*** and which value would be the ***next one dequeued***.

*Find the **two** answers and write them on your answer sheet **correctly labeled**. If using a ScanTron form, write them out to the side of the bubbles, also **correctly labeled**. If not labeled, the order you put your answers will be assumed to be **last value dequeued**, then **next value to be dequeued**.*

Last value dequeued Next value to be dequeued



```
enqueue 3
enqueue 5
enqueue 4
dequeue x
enqueue 7
dequeue x
dequeue x
enqueue 9
```

University Interscholastic League
Computer Science Competition

Number 144 (Invitational B - 2014)

General Directions:

1) **DO NOT OPEN EXAM UNTIL TOLD TO DO SO.**

2) **NO CALCULATOR OF ANY KIND MAY BE USED.**

3) There are 40 questions on this contest exam. You have 45 minutes to complete this contest. If you are in the process of actually writing an answer when the signal to stop is given, you may finish writing that answer.

4) Papers 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 paper until told to do otherwise. Use this time to check your answers.

5) All answers must be written on the answer sheet/Scantron card provided. Indicate your answers in the appropriate blanks provided on the answer sheet or on the Scantron card. Clean erasures are necessary for accurate Scantron grading.

6) You may place as many notations as you desire anywhere on the test paper, but not on the answer sheet or Scantron card, which are reserved for answers only.

7) You may use additional scratch paper provided by the contest director.

8) All questions have ONE and only ONE correct (BEST) answer. There is a penalty for all incorrect answers.

9) A reference to commonly used Java classes is provided at the end of the test, and you may use this reference sheet during the contest. You may detach the reference sheets from the test booklet, but DO NOT DO SO UNTIL THE CONTEST BEGINS.

Scoring:

1) All questions will receive 6 points if answered correctly; no points will be given or subtracted if unanswered; 2 points will be deducted for an incorrect answer.

Note: Correct responses are based on Java, **J2sdk v 1.7.25**, from Sun Microsystems, Inc. All provided code segments are intended to be syntactically correct, unless otherwise stated (i. e. `error` is an answer choice) and any necessary Java 2 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…** ***import static java.lang.System.*;***

QUESTION 1

Which of these is NOT equivalent to $110_2 + 100010_2$?

A. 40_{10} B. 46_8 C. 28_{16} D. 101000_2 E. All are equivalent

QUESTION 2

What is output by the code to the right?

A. 4 B. 4.8

C. 5 D. 5.0

E. There is no output due to a compile error.

```
int h = 24;
h/=5;
out.println(h);
```

QUESTION 3

What is output by the code to the right?

A. 3 B. 4 C. 4.0

D. There is no output due to a compile error.

E. There is no output due to a runtime error.

```
Double [] list = {1.0,2.0,3.0,4};
out.println(list[3]);
```

QUESTION 4

What is output by the code to the right?

A. 369 B. 36912

C. 6912 D. infinite loop

E. There is no output.

```
int k = 3;
do
{
  k+=3;
  out.print(k);
}
while (k!=12);
```

QUESTION 5

What is output by the code to the right?

A. 0 B. 1 C. 5 D. 6 E. 7

```
String s = "beachbum";
out.println(s.indexOf(98,1));
```

QUESTION 6

What is output by the code to the right?

A. 0.12.34.56.7 B. 2.36.74.54.5

C. 4.54.54.54.5 D. 6.74.52.30.1

E. There is no output due to a runtime error.

```
double [] list = {0.1,2.3,4.5,6.7};
list[3]=list[2];
list[1]=list[3];
list[0]=list[1];
for(double d:list)
  out.printf("%.1f",d);
```

QUESTION 7

For which initial values of p and q will this expression output true?

I. `p=false;q=false`
II. `p=false;q=true`
III. `p=true;q=false;`
IV. `p=true;q=true ;`

A. I and IV only
B. II and III only
C. IV only
D. I, II, and III only
E. All will work.

```
boolean p = ?;
boolean q = ?;
out.println(p^q);
```

QUESTION 8

For which of these inputs will the final value of sum be greater than zero?

A. `"a"` B. `"aa"`

C. `"bb"` D. `"cccc"`

E. `""`

```
String s = <string value>;
int sum = 0;
switch(s)
{
  case "a"    : sum += s.length();
  case "bb"   : sum -= s.length();
  case "cccc" : sum *= -s.length();
  case ""     : sum--;
}
out.println(sum);
```

QUESTION 9

What is output by the code to the right?

A. `3.1` B. `5.2` C. `-5.2`

D. `-2.1` E. `-3.1`

```
out.println(Math.min(-5.2,3.1));
```

QUESTION 10

Which statement will correctly output the value 6 from the array shown to the right?

A. `out.print(a[1][2]);` B. `out.print(a[2][3]);`

C. `out.print(a[5]);` D. `out.print(a[2][1]);`

E. `out.print(a[3][2]);`

```
int[][]a={{1,2,3},{4,5,6,7},{8,9}};
```

QUESTION 11

Which of the following correctly replaces **<statement1>** in the Guitar class definition on the right ?

A. `public void`

B. `public int`

C. `private void`

D. `private int`

E. `public static int`

QUESTION 12

Which of the following correctly replaces **<statement2>** in the Guitar class definition on the right ?

A. `( );`

B. `(int n);`

C. `( )`

D. `(String s)`

E. `(int n)`

QUESTION 13

Which of the following correctly replaces **<statement3>** in the Guitar class definition on the right ?

A. `type = s;`

B. `numStrings = n;`

C. `return type;`

D. `return numStrings;`

E. `return 6;`

```
class Guitar
{
  private String type;
  private int numStrings;
  public Guitar()
  {
     type = "acoustic";
     numStrings = 6;
  }
  public Guitar(int n)
  {
     this();
     numStrings = n;
  }
  public Guitar(int n, String s)
  {
     this(n);
     type = s;
  }
  public String toString()
  {
     return type + ": " +
        numStrings + " string";
  }
<statement1>setNumStrings<statement2>
  {
          <statement3>
  }
}
```

QUESTION 14

What is output by the code to the right?

A. 0 B. 15

C. 25 D. 34

E. 1073741823

```
int d = 30;
d = d ^ 15 << 1;
out.println(d);
```

QUESTION 15

What is output by the code to the right?

A. 0 B. 39 C. 40

D. 120 E. 121

```
int j = 0;
do{
  j+=2*j;
  j++;
}
while(j<50);
out.println(j);
```

QUESTION 16

Which term best describes the method type in the code shown to the right?

I. static method
II. void method
III. return method
IV. mutator method

A. I only

D. II only

C. III only

D. I and III only

E. II and IV only

```
static int stuff(int x)
{
  if(x%9>5)
     return (x%9-5);
  if(x%9<5)
     return (x%9+5);
  return (x%9);
}

//client code
out.print(stuff(9));
out.print(stuff(8));
out.print(stuff(14));
```

QUESTION 17

What is output by the client code to the right?

A. -5593680105

B. 439

C. 651

D. 535

E. 9814

QUESTION 18

Which of these statements will return the substring "R"?

A. s.substring(6);
B. s.substring(7);
C. s.substring(6,6);
D. s.substring(6,7);
E. s.substring(7,8);

```
String s = "FenderRumble";
```

QUESTION 19

What is output by the code to the right?

A. -2 B. -3 C. 21 D. 22 E. 25

```
int d = 9;
int f = 60;
int g = 31;
out.println(g-f%d);
```

QUESTION 20

What is output by the code to the right?

A. `000 010 101 111`

B. `000 011 100 111`

C. `001 010 101 110`

D. `001 011 101 110`

E. `000 010 101 110`

```
for(int p = 0; p <= 1; p++)
 for(int q = 0; q <= 1; q++)
  out.print(""+p+q+(p&q|q)+" ");
```

QUESTION 21

What is output by the code to the right?

A. `1` B. `2` C. `3` D. `1.0` E. `1.5`

```
double g = 28.5;
out.println(g%9);
```

QUESTION 22

What is output by the code to the right?

A. `0.79` B. `1.05` C. `1.57` D. `3.14` E. `6.28`

```
d = Math.toRadians(180.0);
out.printf("%.2f\n",d);
```

QUESTION 23

What is output by the code to the right?

A. `2147483644`

B. `-2147483645`

C. `11001111111111111111111111111111` (32 digits)

D. `11000000000000000000000000000000` (32 digits)

E. `1100`

```
int x = 12 << 32;
String s = Integer.toBinaryString(x);
out.println(s);
```

QUESTION 24

What is output by the code to the right?

A. `4 null`

B. `4 6`

C. `5 null`

D. `5 6`

E. There is no output due to a runtime error.

```
ArrayList lost = new ArrayList(5);
lost.add(null);
lost.add(new Integer(6));
lost.add("ball");
lost.add(4.7);
out.println(lost.size()+"
"+lost.get(1));
```

QUESTION 25

Find f(10,5) according to the recursive function definition shown on the right. You may use the space below to do your work.

f(10,5) =

$$f(x,y) = \begin{cases} f(x-y,y-1)+2 & \text{when } x>y \\ x+y & \text{otherwise} \end{cases}$$

A. `5` B. `6` C. `7`

D. `8` E. `10`

QUESTION 26

What is output by the code to the right?

A. `il` B. `vain`

C. `ilovetopaint`

D. There is no output due to a runtime error

E. There is no output due to a compile error

```
String s = "ilovetopaint";
String [] ar = s.split("[pote]");
out.println(ar[1]+ar[5]);
```

QUESTION 27

What is output by the code to the right?

A. `0` B. `5`

C. `100` D. `dead`

E. `walking`

```
String b = (100%5==0)?"walking"
                     :"dead";
out.println(b);
```

QUESTION 28

What is output by the code to the right?

A. `-1` B. `1`

C. `-15` D. `15`

E. `false`

```
s = "SperryRand";
t = "SpecialK";
out.println(s.compareTo(t));
```

QUESTION 29

A. `nine` B. `9`

C. `ten` D. `sepuluh`

E. `null`

```
Map<Integer,String> m =
  new HashMap<Integer,String>();
m.put(10,"ten");
m.put(14,"fourteen");
m.put(9,"nine");
m.put(10,"sepuluh");
out.println(m.get(0));
```

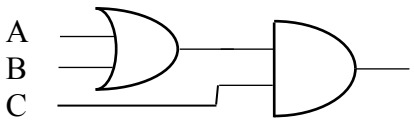
QUESTION 30

Which of the following logical statements is represented by the digital electronics diagram on the right ?

A. A && B || C B. A || B && C

C. A ^ B || C D. (A || B) && C

E. A && B ^ C



QUESTION 31

On the right is a boolean expression using generic notation. Which of the expressions below represents the simplest form of this expression ? (Note : * means AND, + means OR, ⊕ means XOR)

A. $\overline{A} + \overline{B}$ B. $A \oplus B$ C. $A\overline{B} + \overline{A}B$

D. False E. A+B

$(A \oplus B)\ (A + B)$

(this translates to ***"A xor B and A or B)"***)

QUESTION 32

In a typical binary search process, in how many steps will the value 8 be found in the array shown on the right?

A. 3 B. 4

C. 5 D. 6

E. 7

```
0 1 2 3 4 5 6 7 8 9 10 11 12 13
```

QUESTION 33

Which statement below best describes the minimum required <implementation> of class B for the class structure shown on the right?

A. class B is only required to define method **`two()`**.
B. class B is not required to implement anything.
C. class B is required to implement method **`two()`** and override method **`one()`**.
D. class B is only required to override method **`one()`**.
E. This class structure is invalid.

QUESTION 34

Suppose all is correctly defined with this class structure so that method **two()** returns the value 2. What is the output for the client code shown on the right?

A. 0 B. 5

C. 20 D. 40

E. There is no output due to a runtime error.

QUESTION 35

Which of the following is an **INVALID** class B definition?

I.
```
class B extends A{
      int two(){
         return 2;
      }}
```
II.
```
class B implements A{
      x=1;
      int two(){
         return 2;
      }}
```
III.
```
class B extends A{
      int one(){
         return 5;
      }
      int two(){
         return 2;
      }}
```
IV.
```
class B extends A{
      int x = 4;
      int one(){
         return 5;
      }
      int two(){
         return 2;
      }}
```

A. I is invalid
B. II is invalid
C. III is invalid
D. IV is invalid
E. All of these are valid

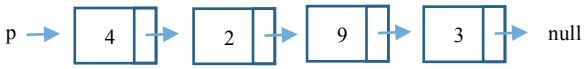
```
abstract class A
{
  int x = 2;
  int one()
  {
     return 5;
  }
  abstract int two();

}
class B extends A
{
  //<implementation>
}

//////////client code///////////////
B bop = new B();
out.println(bop.one()*bop.two()
              *bop.x);
```

QUESTION 36

Suppose a linked list has been implemented as shown in the diagram on the right, with public fields **`data`** and **`next`**. What is the output of the statement below?

```
out.print(p.next.next.data);
```

A. 2 B. 3 C. 4 D. 5 E. 9



QUESTION 37

What is output by the code to the right?

A. `3null`

B. `3false`

C. `3true`

D. `4false`

E. `4true`

```
Set<Integer> sa = new
  TreeSet<Integer>();
sa.add(4);
sa.add(5);
sa.add(4);
sa.add(6);
sa.add(7);
sa.remove(4);
out.print(sa.size());
out.println(sa.contains(6));
```

QUESTION 38

What is the output of this code if the value of ***`<keyboard input>`*** is `3.14`?

A. `Bad data.`

B. `All is good.`

C. `Bad data. All is good.`

D. There is no output.

E. There is no output due to a runtime error.

```
double tx;
try{
  tx = <keyboard input>;
}
catch(Exception ee){
  out.print("Bad data. ");
}
finally{
  out.print("All is good. ");
}
```

QUESTION 39

On the right is a binary tree implementing a max heap, with the 9 in position 0, the 7 in position 1, and the 5 in position 2. The last element added was a 12. In what position does the value 12 settle when the min heap is reestablished in the sifting up process?

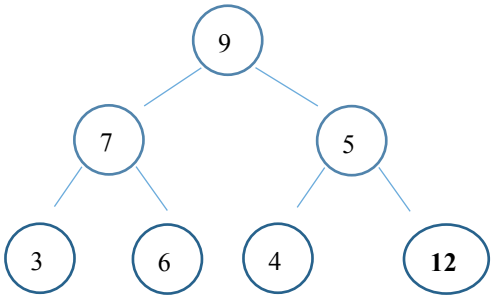
A. position 0

B. position 1

C. position 2

D. position 5

E. position 6

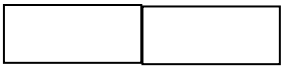


QUESTION 40

OPEN ENDED QUESTION – Using the generic push and pop sequence given on the right (**push** to mean Java's ***enqueue***, **pop** to mean Java's ***dequeue***), process the commands shown on the right into a queue and indicate the ***last value popped*** and which value would be the ***next one popped***.

*Find the **two** answers and write them on your answer sheet **correctly labeled**. If using a ScanTron form, out to the side of the bubbles, also **correctly labeled**. If not labeled, the order you put your answers will be assumed to be **last value popped**, then **next value to be popped**.*

Last value popped Next value to be popped



```
Push 9
Push 7
Pop x
Push 5
Push 8
Push 6
Pop x
Pop x
```

University Interscholastic League
Computer Science Competition

Number 145 (District 1 - 2014)

General Directions:

1) **DO NOT OPEN EXAM UNTIL TOLD TO DO SO.**

2) **NO CALCULATOR OF ANY KIND MAY BE USED.**

3) There are 40 questions on this contest exam. You have 45 minutes to complete this contest. If you are in the process of actually writing an answer when the signal to stop is given, you may finish writing that answer.

4) Papers 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 paper until told to do otherwise. Use this time to check your answers.

5) All answers must be written on the answer sheet/Scantron card provided. Indicate your answers in the appropriate blanks provided on the answer sheet or on the Scantron card. Clean erasures are necessary for accurate Scantron grading.

6) You may place as many notations as you desire anywhere on the test paper, but not on the answer sheet or Scantron card, which are reserved for answers only.

7) You may use additional scratch paper provided by the contest director.

8) All questions have ONE and only ONE correct (BEST) answer. There is a penalty for all incorrect answers.

9) A reference to commonly used Java classes is provided at the end of the test, and you may use this reference sheet during the contest. You may detach the reference sheets from the test booklet, but DO NOT DO SO UNTIL THE CONTEST BEGINS.

Scoring:

1) All questions will receive 6 points if answered correctly; no points will be given or subtracted if unanswered; 2 points will be deducted for an incorrect answer.

Note: Correct responses are based on Java, **J2sdk v 1.7.25**, from Sun Microsystems, Inc. All provided code segments are intended to be syntactically correct, unless otherwise stated (i. e. `error` is an answer choice) and any necessary Java 2 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…** ***import static java.lang.System.*;***

QUESTION 1

Which of these is NOT equivalent to $10101_2 + 10000_2$?

A. 35_{10} B. 45_8 C. 25_{16} D. 100101_2 E. All are equivalent

QUESTION 2

For which initial values of p and q will the code on the right output `true`?

A. `p=true, q=true;` B. `p=false, q=true;`

C. `p=true, q=false;` D. `p=false, q=false;`

E. None of these

```
boolean p=<value1>, q=<value2>;
out.println(p&&q);
```

QUESTION 3

What is output by the code to the right?

A. `4` B. `4.0` C. `5`

D. `5.0` E. `6`

```
double a = 4.1573;
out.println(Math.ceil(a));
```

QUESTION 4

What is output by the code to the right?

A. `13.9` B. `15.7`

C. `27.0` D. `27.4`

E. There is no output due to a compile error.

```
double x = 13.7;
x = 2 * x;
out.println(x);
```

QUESTION 5

What is output by the code to the right?

A. `biminitop biminitop`

B. `biminitop bikinitop`

C. `bikinitop bikinitop`

D. `bikinitop biminitop`

E. There is no output due to a compile error.

```
String s = "biminitop";
String t = s.replace('m','k');
out.println(s+" "+t);
```

QUESTION 6

What is output by the code to the right?

A. `-4.0` B. `-5.0`

C. `-8.2` D. `-9.0`

E. `17.0`

```
out.printf("%.1f\n",9/2-6.5*2);
```

QUESTION 7

What is output by the code to the right?

A. `null` B. `null5` C. `5`

D. There is no output due to a compile error.

E. There is no output due to a runtime error.

```
Integer x = null;
int y = 5;
out.println(x + y);
```

QUESTION 8

What is output by the code to the right?

A. `-50 -2 1` B. `-49 -56 57`

C. `-51 -1 0` D. `-51 -56 55`

E. `-50 -56 56`

```
int x = ~50;
int y = x/7<<3;
int z = ~y;
out.println(x+" "+y+" "+z);
```

QUESTION 9

What is output by the code to the right?

A. `Chill` B. `Dude`

C. `Yo` D. `Sup`

E. `DudeSupWordChill`

```
char a = 'e';
switch(a)
{
  case 'a':out.println("Yo");break;
  case 'e':out.println("Dude");break;
  case 'i':out.println("Sup");break;
  case 'o':out.println("Word");break;
  default :out.println("Chill");
}
```

QUESTION 10

What is output by the code to the right?

A. `7` B. `6`

C. `5` D. `4`

E. There is no output due to a compile error.

```
int x=0;
String [] a = {"red","white","blue"};
char[][]list=new char[a.length][];
for(String s:a)
  list[x]=a[x++].toCharArray();
int k=0;
for(char[]j:list)
  for(char m:j)
     k+="yellow".indexOf(m)>=0?0:1;
  out.println(k);
```

QUESTION 11

The `toString` method is partially implemented in the code to the right. Which statement below would **best** replace <statement1> so that the output in the client code shows `"6 string acoustic"`?

A. `return "6 string acoustic"`

B. `return numStrings + " string " + type`

C. `out.println("6 string acoustic")`

D. `out.println(numStrings + " string " + type)`

E. `"6 string acoustic"`

QUESTION 12

In what Java class is the `toString` method originally defined ?

A. `Guitar`

B. `Object`

C. `System`

D. `String`

E. `Scanner`

QUESTION 13

What term refers to redefining the `toString` method as shown in the code to the right ?

A. inheritance

B. overloading

C. overriding

D. polymorphism

E. interfacing

```
class Guitar
{
  private String type;
  private int numStrings;
  public Guitar()
  {
     type = "acoustic";
     numStrings = 6;
  }
  public Guitar(int n)
  {
     this();
     numStrings = n;
  }
  public Guitar(int n, String s)
  {
     this(n);
     type = s;
  }
  public String toString()
  {
     <statement1>;
  }
////////////////////////////////
////client code
Guitar g = new Guitar();
out.println(g);
```

QUESTION 14

What is output by the code to the right?

A. 523 B. 637

C. 790 D. 951

E. 1003

```
static int stuf(int [] list){
  int k=0,m=0;
  for(int x:list){
     int c=0;
     String s = Integer.toString(x);
     char []ss=s.toCharArray();
     for(char a:ss)
        c+=a-48;
     if(c>k){
        k=c;m=x;
     }
  }
  return m;
}
//client code
int [] list = {523,637,951,790,1003};
out.println(stuf(list));
```

QUESTION 15

What is output by the code to the right?

A. 630 B. 963 C. 9630

D. There is no error, but there is no output

E. There is no output due to a compile error

```
for(int x=9; x==0;x-=3)
  out.print(x);
```

QUESTION 16

What is output by the code to the right?

A. 123456 B. 125456 C. 156456

D. 153456 E. 433456

```
int [] list = {1,2,3,4,5,6};
list[list[1]]=list[list[4]];
list[1]=list[list[3]];
for(int x:list)
 out.print(x);
```

QUESTION 17

What is output by the client code to the right?

A. 2.0 B. 6.0 C. 8.0

D. 10.0 E. 14.0

QUESTION 18

What term best describes the function of the `myst` method defined on the right?

A. Euclid's greatest common factor algorithm

B. Pascal's triangle

C. Leibniz integral rule

D. Newton's law of gravitation

E. Pythagorean theorem

```
public static double myst(double A,
double B)
{
  double AA = Math.pow(A,2);
  double BB = Math.pow(B,2);
  double C = Math.sqrt(AA+BB);
  return C;
}
//client code
double a = 6.0;
double b = 8.0;
out.println(myst(a,b));
```

QUESTION 19

What is output by the code to the right?

A. 1 B. -1 C. 9 D. -9 E. 0

```
String a = "Auburn";
String b = "Alabama";
out.println(a.compareTo(b));
```

QUESTION 20

What is output by the code to the right?

A. B4 114 1110010 B. B4 176 10110000

C. B4 180 10110100 D. B4 B416 1011010000010110

```
String s = "B4";
int i = Integer.parseInt(s,16);
String t = Integer.toBinaryString(i);
out.println(s+" "+i+" "+t);
```

QUESTION 21

What is output by the code to the right?

A. -21 B. -27 C. 21

D. 24 E. 27

```
static int t(int x)
{
  return x%7>3?x-3:x+3;
}
//client code
out.println(t(24));
```

QUESTION 22

What is output by the code to the right?

A. 01100100 B. 0110010 C. 100

D. 1100100 E. 1101100

```
String s;
s = Integer.toBinaryString(100>>32);
out.println(s);
```

QUESTION 23

What is output by the code to the right?

A. -1.00 B. 0.50 C. 0.71

D. 1.00 E. 1.73

```
double d = Math.log(Math.E);
out.printf("%.2f\n",d);
```

QUESTION 24

What is output by the code to the right?

A. 10000000000000000000000000000000 (1, 31 0s)

B. 100000000000000000000000000000000 (1, 32 0s)

C. 11111111111111111111111111111111 (32 1s)

D. 11111111111111111111111111111111 (32 1s, 0)

E. 10000000000000000000000000000001 (1, 30 0s, 1)

```
int x = Integer.MIN_VALUE;
String s = Integer.toBinaryString(x);
out.println(s);
```

QUESTION 25

Find f(-4) according to the recursive function definition shown on the right. You may use the space below to do your work.

f(-4) =

$$f(x,y) = \begin{cases} 2(f(x+2))-f(x+1)+1 & \text{when } x<0 \\ 1 & \text{when } x=0 \\ 0 & \text{when } x>0 \end{cases}$$

A. -6 B. -2 C. 3

D. 5 E. 9

QUESTION 26

What is output by the code to the right?

A. [3, 5, 1, 7] B. [4, 2, 6, 0]

C. [1, 3, 5, 7] D. [0, 2, 4, 6]

E. There is no output due to a compile error.

```
int [] list = {3,4,2,5,1,6,7,0};
ArrayList<Integer> List1 = new
  ArrayList<Integer>();
ArrayList<Integer> List2 = new
  ArrayList<Integer>();
for(int x:list){
     if(x%2==0)
        List1.add(x);
     List2.add(x);
  }
List2.removeAll(List1);
out.println(List1);
```

QUESTION 27

What is output by the code to the right?

A. `I` B. `L`

C. `R` D. `S`

E. `T`

```
String s = "UILDISTRICTCONTEST";
char[]list = s.toCharArray();
int x=1;
PriorityQueue<Character> pq;
pq = new PriorityQueue<Character>();
for(char a:list){
     pq.offer(a);
     if(x%3==0){
        pq.poll();pq.poll();
     }
     x++;
  }
out.println(pq.peek());
```

QUESTION 28

What is output by the code to the right?

A. `5 4.0` B. `4 5.0`

C. `5.0 4.0` D. `4 5`

E. There is no output due to a compile error.

```
int j = 100;
double k = 20;
j/=k;
k/=j;
out.println(j+" "+k);
```

QUESTION 29

What is output by the code to the right?

A. `000 011 101 111` B. `000 011 100 110`

C. `001 011 101 111` D. `000 010 100 111`

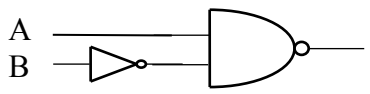
```
for(int p = 0; p <= 1; p++)
  for(int q = 0;q <= 1; q++)
     out.print(""+p+q+(p|q^p)+" ");
```

QUESTION 30

Which of the following logical statements is represented by the digital electronics diagram on the right ?

A. !A || !B B. !(A || !B)

C. !(A && !B) D. !A && !B



QUESTION 31

There is possibly something wrong with the code on the right that would cause a compile error, or it could be just fine. Which answer choice best describes the situation ?

A. There is nothing wrong…the code is fine as is.

B. The interface methods should not have semicolons

C. The class B method A1 needs something inside the {}

D. {} brackets are missing in the interface methods

E. The word `public` needs to precede each method definition.

QUESTION 32

Assuming the code is correct as is, or that the proper fix has been applied so that method A1 outputs the phrase "Hello World" and method A2 returns the value 0, what is the output of the client code listed?

A. `0` B. `HelloWorld0`

C. `HelloWorld`

D. There is no output due to a compile error.

E. There is no output due to a runtime error.

```
interface A
{
  void A1();
  int A2();
}
class B implements A
{
  void A1(){}
  int A2(){return 0;}
}
//client code
A b = new B();
b.A1();
out.print(b.A2());
```

QUESTION 33

What is output by the code to the right?

A. `{a=5, b=7, e=3, f=7}`

B. `{a=5, b=7, c=3, e=3, f=7}`

C. `{c=4, e=3, b=7, a=5, f=7}`

D. `{e=3, b=7, a=5, f=7}`

E. `{a=5, e=3, f=7}`

```
Map<Character,Integer> m = new
  TreeMap<Character,Integer>();
m.put('c',4);
m.put('e',3);
m.put('b',7);
m.put('a',5);
m.put('c',3);
m.put('f',7);
m.remove('c');
out.println(m);
```

QUESTION 34

Which of these is the most efficient O(N) rating?

A. O(N) B. $O(N^2)$ C. O(log N) D. O(N log N) E. O(1)

QUESTION 35

In the code to the right, what value is the last one popped ?

A. 3
B. 5
C. 6
D. 7
E. 9

```
Stack<Integer> s = new
   Stack<Integer>();
s.push(3);
s.push(5);
s.push(9);
s.pop();
s.push(6);
s.pop();
s.pop();
s.push(2);
s.push(7);
```

QUESTION 36

If A and B are Boolean values, which is the most simplified expression for A*0 + B + 1, where * means AND, + means OR, 0 means false, and 1 means true?

A. `0` B. `1` C. `A` D. `B` E. `A+B`

QUESTION 37

What is the length of the longest diagonal of 1s printed by this code?

A. 3
B. 4
C. 5
D. 7
E. 6

```
for(int x=0;x<8;x++) {
  for(int y=0;y<8;y++)
     out.print(((x+y)%4==0)?1:0);
     out.println();
  }
```

QUESTION 38

What is output by the code to the right?

A. `9` B. `10` C. `11`

D. `12` E. `16`

```
int a = 45;
int b = 34;
out.println(a%10+b/10+b%10);
```

QUESTION 39

In graph 1 on the right, the adjacency matrix would look like this, where 1 means a one way connection and 0 would mean no connection:

	A	B	C
A	0	1	1
B	1	1	0
C	0	0	0

Which choice below represents the adjacency matrix for Graph 2 on the right?

A.

	A	B	C	D
A	0	1	1	0
B	1	0	0	1
C	0	0	1	0
D	0	1	0	0

B.

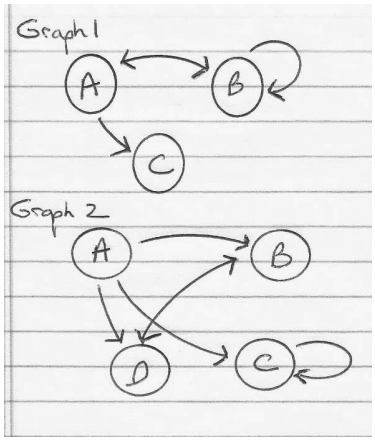
	A	B	C	D
A	0	1	1	1
B	0	0	0	1
C	0	0	0	0
D	0	1	0	0

C.

	A	B	C	D
A	0	1	1	1
B	0	0	0	1
C	0	0	1	0
D	0	1	0	0

D.

	A	B	C	D
A	0	0	0	0
B	1	0	0	1
C	1	0	1	0
D	1	1	0	0



QUESTION 40

What is output by the code to the right?

A. `16.0 15.0`

B. `16.0 16.0`

C. `14.0 17.0`

D. `12.0 18.0`

E. `5.0 20.0`

```
double a = 5,b=20;
do{
  if (a<b)
     a=a+(int)(b/a)+1;
  b=b-1;
  }
while(a<=b);
out.println(a+" "+b);
```

University Interscholastic League
Computer Science Competition

Number 146 (District 2 - 2014)

General Directions:

1) **DO NOT OPEN EXAM UNTIL TOLD TO DO SO.**

2) **NO CALCULATOR OF ANY KIND MAY BE USED.**

3) There are 40 questions on this contest exam. You have 45 minutes to complete this contest. If you are in the process of actually writing an answer when the signal to stop is given, you may finish writing that answer.

4) Papers 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 paper until told to do otherwise. Use this time to check your answers.

5) All answers must be written on the answer sheet/Scantron card provided. Indicate your answers in the appropriate blanks provided on the answer sheet or on the Scantron card. Clean erasures are necessary for accurate Scantron grading.

6) You may place as many notations as you desire anywhere on the test paper, but not on the answer sheet or Scantron card, which are reserved for answers only.

7) You may use additional scratch paper provided by the contest director.

8) All questions have ONE and only ONE correct (BEST) answer. There is a penalty for all incorrect answers.

9) A reference to commonly used Java classes is provided at the end of the test, and you may use this reference sheet during the contest. You may detach the reference sheets from the test booklet, but DO NOT DO SO UNTIL THE CONTEST BEGINS.

Scoring:

1) All questions will receive 6 points if answered correctly; no points will be given or subtracted if unanswered; 2 points will be deducted for an incorrect answer.

Note: Correct responses are based on Java, **J2sdk v 1.7.25**, from Sun Microsystems, Inc. All provided code segments are intended to be syntactically correct, unless otherwise stated (i. e. `error` is an answer choice) and any necessary Java 2 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…** ***import static java.lang.System.*;***

QUESTION 1

Which of these is NOT equivalent to $100010_2 + 100000_2$?

A. 66_{10} B. 112_8 C. 42_{16} D. 1000010_2 E. All are equivalent

QUESTION 2

What is output by the code to the right?

A. `5 19.0 3.8` B. `5 19.0 3`

C. `5.0 19.0 3.8` D. `5 19 4`

E. There is no output due to a compile error.

```
int w = 5;
double z = 19;
double q = z/w;
out.println(w+" "+z+" "+q);
```

QUESTION 3

What is output by the code to the right?

A. `falsefalsefalse` B. `truefalsefalse`

C. `truefalsetrue` D. `truetruetrue`

E. `truetruefalse`

```
Integer x = 5;
Integer y = x;
out.print(x==y);
y = 5;
out.print(x==y);
y = new Integer(5);
out.println(x==y);
```

QUESTION 4

What is output by the code to the right?

A. `5` B. `6 7 8`

C. `5 6 7 8` D. `5 6 7`

E. There is no output.

```
int x = 5;
while (x<=7)
  out.print(x+++" ");
```

QUESTION 5

What is output by the code to the right?

A. `1` B. `2` C. `5`

D. `8` E. `9`

```
String s = "bassGuitar";
out.println(s.lastIndexOf("a"));
```

QUESTION 6

What is output by the code to the right?

A. `5` B. `6` C. `9`

D. `10` E. `11`

```
int list[] = {1,3,5,2,4};
out.println(list[1]+list[3]);
```

QUESTION 7

For which initial values of p and q will this expression output `false`?

A. `true true` B. `true false`

C. `false true` D. `false false`

```
boolean p = <value1>,q = <value2>;
out.println(p||q);
```

QUESTION 8

What is output by the code to the right?

A. `00` B. `0`

C. `06` D. `66`

E. `60`

```
int z = 42;
if(z%7==0)
  out.print(z/7);
  out.println(z%7);
```

QUESTION 9

What is output by the code to the right?

A. `63 254` B. `254 63` C. `63 -2`

D. `508 31` E. `31 508`

```
int b = 127;
int c = 127;
out.println((b>>=2)+" "+(c<<=2));
```

QUESTION 10

What is output by the code to the right?

A. `-5.0` B. `-6.0`

C. `5.0` D. `6.0`

E. There is no output due to a compile error.

```
double f = -5.9423;
out.println(Math.floor(f));
```

QUESTION 11

Which statements would correctly replace <statement1> in the client code on the right to correctly modify the current Guitar object g into a 5 string bass guitar?

I. `g.getNumStrings(5);`
II. `g.setNumStrings(5);`
III. `g = new Guitar(5) ;`
IV. `g = new Guitar(5,"bass") ;`

A. I only

B. II only

C. III only

D. III and IV only

E. II, III, and IV only

QUESTION 12

Which statement would correctly replace `<statement2>` in the client code shown to output the type for the Guitar object g?

A. `out.println(g.getType());`

B. `out.println(g.setType("bass"));`

C. `out.println(g.getNumStrings());`

D. `out.println(g.setNumStrings(4));`

E. `out.println(g);`

QUESTION 13

Assuming the statements above have been correctly defined as described what is the output of the client code?

A. `4 string bass`

B. `5 string bass`

C. `bass: 4 string`

D. `bass: 5 string`

E. `6 string acoustic`

```
static class Guitar
{
  private String type;
  private int numStrings;
  public Guitar(){
     type = "acoustic";
     numStrings = 6;
  }
  public Guitar(int n){
     this();
     numStrings = n;
  }
  public Guitar(int n, String s){
     this(n);
     type = s;
  }
  public void setType(String s){
     type = s;
  }
  public String getType(){
     return type;
  }
  public void setNumStrings(int n){
     numStrings = n;
  }
  public int getNumStrings(){
     return numStrings;
  }
  public String toString()
  {
     return type+": "+numStrings+
     " string";
  }
////////////////////////////////
////client code
Guitar g = new Guitar(4,"bass");
<statement1>
<statement2>
out.println(g);
```

QUESTION 14

What is output by the code to the right?

A. `5` B. `5.6`

C. `7` D. `7.3`

E. `9`

```
out.printf("%.1f\n",3*4.2-7);
```

QUESTION 15

What is output by the code to the right?

A. `abcdef` B. `defabc`

C. `cbafed` D. `fedcba`

E. There is no output due to a compile error

```
static void showGrid(char[][]g){
 for(int r=g.length-1;r>=0;r--){
  for(int c=g[0].length-1;c>=0;c--)
   out.print(g[r][c]);
 }
}
//client code
char[][]g={{'a','b','c'},
           {'d','e','f'}};
showGrid(g);
```

QUESTION 16

For which of these input values will the output be 9?

A. `240` B. `100` C. `600`

D. `250` E. `260`

```
double d = <input>;
int x=0;
do{
   d/=2;
   x++;
}while(d>=1.0);
out.println(x);
```

QUESTION 17

What value is in position 4 after the client code to the right executes?

A. `6` B. `-1` C. `2`

D. `5` E. `4`

QUESTION 18

What is the greatest value in the list after the method call?

A. `0` B. `-1` C. `2`

D. `5` E. `4`

```
public static void Myst(int[]list){
  for(int j = 3;j<=5;j++)
     list[j]=list[j-2]-list[j-1];
}
//client code
int [] list = new int[6];
list[1]=5;
list[2]=2;
Myst(list);
```

QUESTION 19

Which of these choices could replace **<statement1>** to output the value 5?

I. substring(15)
II. substring(16)
III. substring(5,10)
IV. substring(7,12)
V. substring(10,16)

A. I only B. I, II, and III only

C. I, III, and IV only D. II, III, and IV only

E. All will work correctly to output the value 5

```
String a = "01234567890123456789";
out.println(a.<statement1>.length());
```

QUESTION 20

What is output by the code to the right?

A. `000 010 101 110` B. `000 010 101 111`

C. `001 011 101 110` D. `001 010 100 111`

```
for(int p = 0; p <= 1; p++)
  for(int q = 0;q <= 1; q++)
     out.print(""+p+q+(p^q&p)+" ");
```

QUESTION 21

What is output by the code to the right?

A. `2.0` B. `3.0` C. `4.0`

D. `5.0` E. `6.0`

```
double y = 42;
y %= 13;
y = ++y;
out.println(y);
```

QUESTION 22

What is output by the code to the right?

A. `1010` B. `1100` C. `10`

D. `00001010` E. `1110`

```
String s=Integer.toBinaryString(10);
out.println(s);
```

QUESTION 23

What is output by the code to the right?

A. 64 B. 8 C. 4

D. 32 E. 16

```
out.println(Short.SIZE);
```

QUESTION 24

What is output by the code to the right?

A. -1.00 B. 0.00 C. 1.00

D. 2.72 E. 3.14

```
double d = Math.log(Math.E);
out.printf("%.2f",d);
```

QUESTION 25

Find f(6,5) according to the recursive function definition shown on the right. You may use the space below to do your work.

f(6,5) =

$$f(x,y) = \begin{cases} 2+f(x-3,y-1) & \text{when } x>y \text{ and } x>0 \\ 1+f(y,x) & \text{when } y>=x \text{ and } x>0 \\ 0 & \text{when } x<=0 \end{cases}$$

A. 8 B. 6 C. 5

D. 2 E. 3

QUESTION 26

What is output by the code to the right?

A. 32 B. 31

C. 1 D. 2

E. There is no output due to a compile error.

```
int x = -1>>>32;
String s = Integer.toBinaryString(x);
out.println(s.length());
```

QUESTION 27

What is output by the code to the right?

A. 3a B. 4a C. 4ade

D. 3vea E. 4IL

```
String s = "ILoveAParade";
String []a = s.split("[j-rM-Q]");
List<String> b = Arrays.asList(a);
out.println(b.size()+b.get(3));
```

QUESTION 28

What is output by the code to the right?

A. 34 114 B. 34 44

C. 44 54 D. 35 45

E. 34 54

```
Integer i = 34;
String s = i.toString();
int x = i;
String t = Integer.toString (x,5);
out.println(s+" "+t);
```

QUESTION 29

What is output by the code to the right?

A. winterwind winterwind

B. winterwind wimterwimd

C. suntersund sumtersumd

D. winterwind suntersund

E. winterwimd sumtersumd

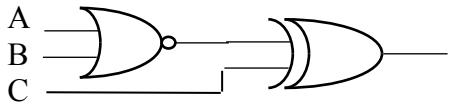
```
String w = "winterwind";
w.replaceAll("win","sun");
String s = w.replace('n','m');
out.println(w+" "+s);
```

QUESTION 30

Which of the following logical statements is represented by the digital electronics diagram on the right ?

A. A ^ B || C B. !(A || B) ^ C

C. !(A ^ B) || C D. A || B ^ C



QUESTION 31

There is possibly something wrong with the code on the right that would cause a compile error, or it could be just fine. Which answer choice best describes the situation ?

A. There is nothing wrong…the code is fine as is.

B. The abstract class methods should not have semicolons

C. The word `extends` should be `implements` instead

D. {} brackets are missing in the abstract class methods

E. The word `public` needs to precede each method definition.

```
abstract class A{
  abstract void A1();
  abstract int A2();
}

class B extends A{
  void A1(){}
  int A2(){return 0;}
}
//client code
A b = new B();
b.A1();
out.print(b.A2());
```

QUESTION 32

Assuming the code is updated so that method A1 outputs the phrase "`I made a `" and method A2 returns the value `240`, what is the output of the client code listed?

A. `0` B. `240`

C. `I made a 240`

D. There is no output due to a compile error.

E. There is no output due to a runtime error.

QUESTION 33

What is output by the code to the right?

A. `3` B. `7` C. `9`

D. `5` E. `6`

```
Queue<Integer> q = new
        LinkedList<Integer>();
q.add(3);q.add(5);q.add(9);
q.poll();q.add(6);q.poll();
q.poll();q.add(2);q.add(7);
out.println(q.peek());
```

QUESTION 34

Which of these is the least efficient O(N) rating?

A. O(N) B. $O(N^2)$ C. O(log N) D. O(N log N) E. O(1)

QUESTION 35

What is output by the code to the right?

A. `16` B. `15` C. `14`

D. `13` E. `12`

```
String ss="Now is the time for all"+
  " good men to come to the aid of"+
  " their country";
String [] a = ss.split(" ");
Set<String> s = new
   HashSet<String>(Arrays.asList(a));
out.println(s.size());
```

QUESTION 36

If A and B are Boolean values, which is the most simplified expression for A*B*A+0, where * means AND, + means OR, 0 means false, and 1 means true?

A. `0` B. `1` C. `A` D. `A*A*B` E. `A*B`

QUESTION 37

What bottom-left-corner to top-right-corner diagonal series of characters is produced by this code??

A. `abcde` B. `DEFGH`
C. `ABCDE` D. `defgh`
E. `01234`

```
for(int x=0;x<5;x++){
 for(int y=0;y<5;y++)
  out.print(((x+y)%5==4)
        ?(char)(y+100):'-');
  out.println();
}
```

QUESTION 38

What is output by the code to the right?

A. `45657` B. `4565` C. `5657`
D. `7565` E. `5654`

```
LinkedList<Integer>a = new
LinkedList<Integer>();
a.push(4); a.add(5);
a.offer(6);a.add(3,5);
a.offerLast(7);a.pollFirst();
Iterator<Integer> i =
    a.descendingIterator();
while(i.hasNext())
          out.print(i.next());
```

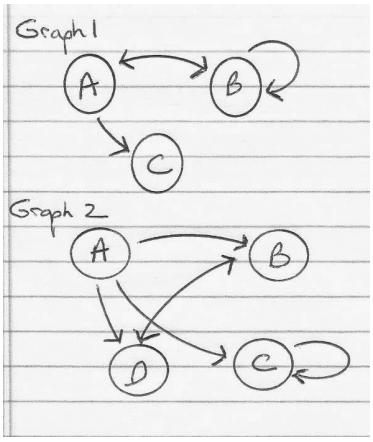
QUESTION 39

In graph 1 on the right, the adjacency matrix would look like this, where 1 means a one way connection and 0 would mean no connection:

	A	B	C
A	0	1	1
B	1	1	0
C	0	0	0

How many zeroes would be in the adjacency matrix for Graph 2?

A. `6` B. `10` C. `16`
D. `13` E. `3`



QUESTION 40

What is output by the code to the right?

A. `10 10` B. `10 20`
C. `10 25` D. `25 10`
E. `20 20`

```
static void p(int []a,int []b){
  a[0]=a[0]+b[0];
  b[0]=a[0]-b[0];
  a=b;
}
//client code
int [] x={10};
int [] y={5};
p(x,y);
p(y,x);
out.println(x[0]+" "+y[0]);
```

No Test Material On This Page

University Interscholastic League
Computer Science Competition

Number 147 (Region - 2014)

General Directions:

1) **DO NOT OPEN EXAM UNTIL TOLD TO DO SO.**

2) **NO CALCULATOR OF ANY KIND MAY BE USED.**

3) There are 40 questions on this contest exam. You have 45 minutes to complete this contest. If you are in the process of actually writing an answer when the signal to stop is given, you may finish writing that answer.

4) Papers 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 paper until told to do otherwise. Use this time to check your answers.

5) All answers must be written on the answer sheet/Scantron card provided. Indicate your answers in the appropriate blanks provided on the answer sheet or on the Scantron card. Clean erasures are necessary for accurate Scantron grading.

6) You may place as many notations as you desire anywhere on the test paper, but not on the answer sheet or Scantron card, which are reserved for answers only.

7) You may use additional scratch paper provided by the contest director.

8) All questions have ONE and only ONE correct (BEST) answer. There is a penalty for all incorrect answers.

9) A reference to commonly used Java classes is provided at the end of the test, and you may use this reference sheet during the contest. You may detach the reference sheets from the test booklet, but DO NOT DO SO UNTIL THE CONTEST BEGINS.

Scoring:

1) All questions will receive 6 points if answered correctly; no points will be given or subtracted if unanswered; 2 points will be deducted for an incorrect answer.

Note: Correct responses are based on Java, **J2sdk v 1.7.25**, from Sun Microsystems, Inc. All provided code segments are intended to be syntactically correct, unless otherwise stated (i. e. `error` is an answer choice) and any necessary Java 2 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…** ***import static java.lang.System.*;***

QUESTION 1

Which of these is NOT equivalent to 111011110_2 - $F2_{16}$?

A. 234_{10} B. 354_8 C. EC_{16} D. 11101100_2 E. All are equivalent

QUESTION 2

What is output by the code to the right?

A. `0.2` B. `0.6` C. `6.2`

D. `7` E. `7.8`

```
out.println(16 % 9 - 4 * 0.2);
```

QUESTION 3

What is output by the code to the right?

A.
```
Hello
Goodbye
4_
true
```

B.
```
HelloGoodbye
4_true
```

C. `Hello Goodbye 4_true`

D. There is no output due to a compile error.

E. There is no output due to a runtime error.

```
out.print("Hello");
out.println("Goodbye");
out.printf("%s_%s\n",4,true);
```

QUESTION 4

What is output by the code to the right?

A. `balloonBomb` B. `balloonbomb`

C. `Balloonbomb` D. `BalloonBomb`

E. There is no output due to a compile error.

```
String s = "BalloonBomb";
s.toLowerCase();
out.println(s);
```

QUESTION 5

For which initial values of p and q will the code on the right output `false`?

A. `p=true, q=true;` B. `p=false, q=true;`

C. `p=true, q=false;` D. `p=false, q=false;`

E. None of these

```
boolean p=<value1>, q=<value2>;
out.println(p|!p&q);
```

QUESTION 6

Which of the following would most accurately replace `<datatype>` in the following expression?

A. `float` B. `double`

C. `char` D. `int` E. `long`

```
<datatype> num = Math.round(3.4);
```

QUESTION 7

What is output by the code to the right?

A. `25.0` B. `25.1`

C. `28.0` D. `28.2`

E. There is no output due to a compile error.

```
int a = 70;
double b = 5.2;
b += a /= 3;
out.println(b);
```

QUESTION 8

What is output by the code to the right?

A. `B==b` B. `B!=b`

C. There is no error, but there is no output.

D. There is no output due to a compile error.

E. There is no output due to a runtime error.

```
char a = 'B';
char b = 'b';
if(a==b)
  out.println(a+"=="+b);
else
  out.println(a+"!="+b);
```

QUESTION 9

If c stands for column in the code to the right, what column would contain water when first detected?

A. column 3 B. column 5

C. column 15 D. column 14

E. It is not possible to determine this.

```
int c = 0;
boolean isWaterAhead = false;
while(!isWaterAhead)
{
  if((c+1)%5==0&&(c+1)%3==0)
     isWaterAhead=true;
  c++;
}
```

QUESTION 10

What is output by the code to the right?

A. `65` B. `57`

C. `129` D. `115` E. `147`

```
char[]list = new char[4];
list[1]=65;
list[3]=50;
int sum=list[1]+list[2]+list[3];
out.println(sum);
```

QUESTION 11

Which of these Java classes can be used in the input process, either from keyboard or from file?

I. File II. FileWriter III. Scanner IV. PrintWriter

A. I only B. I and II only C. I and III only D. III only

E. All of these can be used for input

QUESTION 12

What is output by the code to the right?

A. `4 4.8` B. `6 115.20`

C. `13 9.6` D. `32 19.2`

E. `70 38.4`

```
int a = 0;
double b = 2.4;
while (a+b<25){
     b*=2;a+=b;
  }
out.println(a+" "+b);
```

QUESTION 13

What is output by the code to the right?

A. `40` B. `21`

C. `15` D. `11` E. `10`

```
int f = 5;
int g = f<<2+1;
out.println(g);
```

QUESTION 14

What is output by the code to the right?

A. `4` B. `8` C. `16`

D. `32` E. `64`

```
out.println(Float.SIZE);
```

QUESTION 15

What is output by the code to the right?

A. `[4, 1]` B. `[4, 1, 2]`

C. `[4, 1, 3]` D. `[4, 3, 2]`

E. There is no output due to a compile error

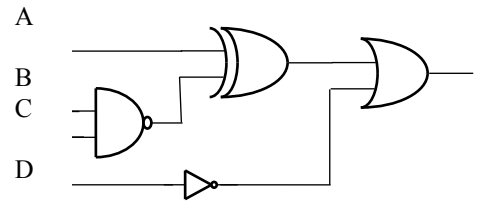
```
ArrayList<Integer> list = new
    ArrayList<Integer>();
list.add(4);
list.add(1);
list.add(3);
list.add(2);
list.remove(2);
out.println(list);
```

QUESTION 16

Which of the following logical statements is represented by the digital electronics diagram on the right ?

A. A^!(B&C)|!D B. A^(B&C)|D

C. A^ !(B|C)&!D D. A|!(B&C)^!D



QUESTION 17

How many times will the word **red** be output by the code to the right?

A. 36 B. 35 C. 26

D. 25 E. 24

```
for(int a=45;a<=80;a++)
out.print((a>50? a<75? "red":
          "green": "blue"));
```

QUESTION 18

What is output by the code to the right?

A. falsefalse B. falsetrue

C. truefalse D. truetrue

E. There is no output due to a compile error

```
String s = "UILRegion2014";
boolean p,q;
p = s.matches(".[^WIN]+.*");
q = s.matches(".............");
out.println(""+p+q);
```

QUESTION 19

What is output by **statement 1** in the code to the right?

A. The dog is a: dachshund

B. The animal is a: dachshund

C. The dog is a: mammal

D. The animal is a: mammal

E. There is no output.

QUESTION 20

What is output by **statement 2** in the code to the right?

A. The dog is a: dachshund

B. The dog is a: mammal

C. The type is: dachshund

D. The type is: mammal

E. There is no output.

QUESTION 21

Which of the statements below is most accurate?

A. In an inheritance situation as shown in the code to the right, early (static) binding occurs at run time, while late (dynamic) binding occurs at compile time.

B. In the client code to the right, the **show()** method that is called is the one that belongs to the Animal class.

C. In the client code to the right, the **type** instance variable that is used is the one that belongs to the Animal class.

D. All of the above statements are true.

E. None of these statements are true.

```
class Animal
{
  public String type = "mammal";
  public void show()
  {
     out.println("The animal is a: "
                 + type);
  }
}
class Dog extends Animal
{
  public String type;
  public Dog(String type)
  {
     this.type = type;
  }
  public void show()
  {
     out.println("The dog is a: "
                  + type);
  }
}
//client code
Animal doggie = new Dog("dachshund");
//statement 1
doggie.show();
//statement 2
out.println("The type is: "
            + doggie.type);
```

QUESTION 22

What is output by the code to the right?

A. 7 2 4 3 8 1 -24130- 1 2 3 4 7 8

B. 7 2 4 3 8 1 -24103- 1 2 3 4 7 8

C. 7 2 4 3 8 1 -21034- 1 2 3 4 7 8

D. 7 2 4 3 8 1 -21043- 1 2 3 4 7 8

E. 7 2 4 3 8 1 -21403- 1 2 3 4 7 8

QUESTION 23

Which algorithm process listed below best describes the code to the right?

A. Insertion sort

B. Quick sort

C. Merge sort

D. Bubble sort

E. Heap sort

QUESTION 24

What is the most restrictive bound on the runtime of this process, where N represents the number of items in list?

A. O(1)

B. O(N)

C. O(log N)

D. O(N log N)

E. O(N^2)

```
class SomeSort {
  static int[] numbers;
  static int[] helper;
public static void sort(int[] values)
{
    numbers = values;
    int number = values.length;
    helper = new int[number];
    somesort(0, number - 1);
}

public static void somesort(int low,
                            int high)
{
    if (low < high) {
      int middle=low+(high-low)/2;
      out.print (middle);
      somesort(low, middle);
      somesort(middle + 1, high);
      combine(low, middle, high);
    }
  }
public static void combine(int low,
               int middle, int high)
{
  for (int i = low; i <= high; i++) {
      helper[i] = numbers[i];
    }
  int i = low;
  int j = middle + 1;
  int k = low;
  while (i <= middle && j <= high) {
      if (helper[i] <= helper[j]) {
        numbers[k] = helper[i];
        i++;
      } else {
        numbers[k] = helper[j];
        j++;
      }
      k++;
    }
    while (i <= middle) {
      numbers[k] = helper[i];
      k++;
      i++;
    }
  }
}

//client code
int [] list = {7,2,4,3,8,1};
  for(int x:list)
     out.print(x+" ");
  out.print("-");
  SomeSort.sort(list);
  out.print("- ");
  for(int x:list)
     out.print(x+" ");
```

QUESTION 25

What is output by the code to the right?

A. 16 B. 24

C. 30 D. 36

E. There is no output due to a compile error

```
int [][] g = new int [5][];

for(int x=0;x<g.length;x++)
   g[x]=new int[(x+1)*2];
int d=0;

for(int r=0;r<g.length;r++)
   for(int c=0;c<g[r].length;c++)
      d++;

out.println(d);
```

QUESTION 26

What is output by the code to the right?

A. 5 B. 10 C. 54

D. 135 E. 525

```
out.println(Integer.toString(75,7));
```

QUESTION 27

What is output by the code to the right?

A. 12VLein

B. 24LUino

C. 24ILonU

D. 2L4ino

E. einVL21

```
String s = "Region UIL 2014";
char[]list = s.toCharArray();
PriorityQueue<Character> pq =
    new PriorityQueue<Character>();

for(int x = 0;x<list.length;x++)
{
  pq.add(list[x]);
  if(x%2==0)
     pq.remove();
}
for(char a:pq)
  out.print(a);
```

QUESTION 28

What is output by the code to the right?

A. 3 B. 7

C. 9 D. 14 E. 15

```
int a=13,b=4,c=7,d=6;
out.println(a^b&c|d);
```

QUESTION 29

What is output by the code to the right?

A. 001 010 101 110 B. 001 011 101 111

C. 000 010 100 110 D. 001 011 101 110

E. 000 010 101 110

```
for(int p = 0; p <= 1; p++)
  for(int q = 0;q <= 1; q++)
  {
     boolean P = p==1;
     boolean Q = q==1;
     boolean R = !(P|Q)|(P&!Q);
     int r = R?1:0;
     out.print(""+p+q+r+" ");
  }
```

QUESTION 30

What is output by the code to the right?

A. 0 B. 100

C. -245 D. 172 E. -2940

```
int a = (int)Math.pow(14,2);
int b = a>>4;
int c = b%5;
out.println(a-b*c);
```

QUESTION 31

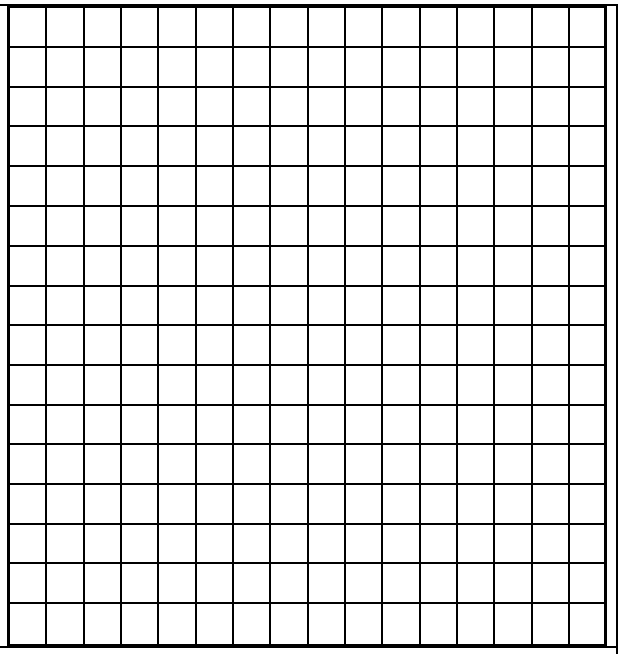
Consider the following recursive algorithm for painting a square:

1. Given a square with side length 16 feet
2. If the length of a side is equal to 1, stop the process for that square, otherwise continue.
3. Divide the square into 4 equal size squares.
4. Paint one of the these 4 smaller squares.
5. Return step 2 for each of the remaining 3 squares.

How many square feet of this square will be painted?

A. 256 B. 148 C. 175

D. 81 E. 64



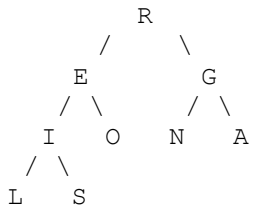
QUESTION 32

Which of the following is the preorder traversal of the tree shown to the right?

A. REILSONGA B. REILSOGNA

C. AEGILNORS D. LISEORNGA

E. LSIOENAGR



QUESTION 33

What is output by the code to the right?

A. [5, 1, 3, 6, 7]

B. [1, 3, 4, 5, 6, 7]

C. [4, 1, 3, 5, 6, 7]

D. [4, 5, 1, 3, 5, 6, 7]

E. [4, 5, 1, 3, 6, 7]

```
LinkedList<Integer>a = new
     LinkedList<Integer>();
a.offerFirst(4);
a.addLast(5);
a.element();
int [] list = {5,3,6,7,1};
Set<Integer> set = new
     TreeSet<Integer>();
for(int x:list)
     set.add(x);
a.addAll(set);
a.removeLastOccurrence(5);
out.println(a);
```

QUESTION 34

In the methods of the Queue interface, three pairs of methods are similar, with **`add()`** and **`offer()`** both inserting an element into the queue, **`peek()`** and **`element()`** both returning the head value of the queue without removing it, and **`poll()`** and **`remove()`** both returning and removing the head value. So what, if any, is the difference between each pair of methods? Select the statement below that accurately describes this situation.

A. **`add()`** throws an exception if the queue is full; **`offer()`** returns false without throwing an exception

B. **`peek()`** returns **`null`** if the queue is empty; **`element()`** throws an exception

C. **`poll()`** returns **`null`** if the queue is empty; **`remove()`** throws an exception

D. All of these statements are false
E. All of these statements are true

QUESTION 35

On the right is the definition of a Boat class. How many constructors are there in this definition ?

A. 1
B. 2
C. 3
D. 4
E. 5

QUESTION 36

What is the output of the statement 1 in the client code below?

A. `skiff 1 1.5`
B. `"skiff" 1 1.5`
C. `1 1.5 skiff`
D. `1-hull "skiff": minimum 1.5 feet of water`
E. `1-hull skiff: minimum 1.5 feet of water`

QUESTION 37

Which of the following would most directly replace **`code segment 2`** to produce the output :

```
2-hull skiff: minimum 2.0 feet of water
```

I.
```
Boat b2 = new Boat(2);
b2.setDraft(2);
out.println(b2);
```

II.
```
Boat b2 = new Boat(2.0);
b2.setNumHulls(2);
out.println(b2);
```

III.
```
Boat b2 = new Boat("catamaran");
b2.setDraft(2);
b2.setNumHulls(2.0);
b2.setType("skiff");
out.println(b2);
```

A. I only
B. II only
C. III only
D. None of these
E. All of these

```
static class Boat
{
 private String type;
 private int numHulls;
 private double draft;

 public Boat(){
  type = "skiff";  numHulls = 1;
  draft = 1.5;
 }
 public Boat(int n){
  type = "skiff";numHulls = n;
  draft = 1.5;
 }
 public Boat(double n) {
  type = "skiff";numHulls = 1;
  draft = n;
 }
 public Boat(String n){
  type = n; numHulls = 1;
  draft = 1.5;
 }
 public void setType(String s){
  type = s;
 }
 public String getType(){
  return type;
 }
 public void setDraft(double d){
  draft = d;
 }

 public double getDraft(){
  return draft;
 }

 public void setNumHulls(int n){
  numHulls = n;
 }

 public int getNumHulls(){
  return numHulls;
 }

 public String toString(){
  return numHulls+"-hull "+type
         +": minimum "+draft
         +" feet of water";
 }
}

//client code
Boat b1 = new Boat();

//statement 1
out.println(b1);

//code segment 2
```

QUESTION 38

What is output by **`statement 1`** in the code to the right?

A. `65` B. `-65`

C. `97` D. `-97`

E. There is no output due to a compile error

QUESTION 39

What is output by **`statement 2`** in the code to the right?

A. `94` B. `-94`

C. `62` D. `-62`

E. There is no output due to a compile error

```
public static int myst(char[]list)
{
   int s = 0;
   for(char a:list)
          s+=a%2==0?a:-a;
   return s;
}
//client code
//statement 1
out.print(myst("bed".toCharArray()));



//statement 2
out.print(myst("MET".toCharArray()));
```

QUESTION 40

After the following elements {7, 2, 4, 9, 5, 6, 1} are correctly inserted into a min heap, which element is the right child of the root?

A. `2`

B. `4`

C. `5`

D. `6`

E. `7`

No Test Material On This Page

University Interscholastic League
Computer Science Competition

Number 148 (State - 2014)

General Directions:

1) **DO NOT OPEN EXAM UNTIL TOLD TO DO SO.**

2) **NO CALCULATOR OF ANY KIND MAY BE USED.**

3) There are 40 questions on this contest exam. You have 45 minutes to complete this contest. If you are in the process of actually writing an answer when the signal to stop is given, you may finish writing that answer.

4) Papers 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 paper until told to do otherwise. Use this time to check your answers.

5) All answers must be written on the answer sheet/Scantron card provided. Indicate your answers in the appropriate blanks provided on the answer sheet or on the Scantron card. Clean erasures are necessary for accurate Scantron grading.

6) You may place as many notations as you desire anywhere on the test paper, but not on the answer sheet or Scantron card, which are reserved for answers only.

7) You may use additional scratch paper provided by the contest director.

8) All questions have ONE and only ONE correct (BEST) answer. There is a penalty for all incorrect answers.

9) A reference to commonly used Java classes is provided at the end of the test, and you may use this reference sheet during the contest. You may detach the reference sheets from the test booklet, but DO NOT DO SO UNTIL THE CONTEST BEGINS.

Scoring:

1) All questions will receive 6 points if answered correctly; no points will be given or subtracted if unanswered; 2 points will be deducted for an incorrect answer.

Note: Correct responses are based on Java, **J2sdk v 1.7.25**, from Sun Microsystems, Inc. All provided code segments are intended to be syntactically correct, unless otherwise stated (i. e. `error` is an answer choice) and any necessary Java 2 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…** ***import static java.lang.System.*;***

QUESTION 1

Which of these is NOT equivalent to $527_8 + 910_{10}$?

A. 1253_{10} B. 2345_8 C. $4E5_{16}$ D. 1001100101_2 E. All are equivalent

QUESTION 2

What is output by the code to the right?

A. `1.7` B. `2.4` C. `5.2`
D. `5.4` E. `7.5`

```
out.println(23 / 4 + 9.4 % 3);
```

QUESTION 3

What is output by the code to the right?

A. `Atrue`
B. `true`
C. `falseA`
D. There is no output due to a compile error.
E. There is no output due to a runtime error.

```
out.printf("%s%s",false,'A',"true");
```

QUESTION 4

What is output by the code to the right?

A. `falsefalse` B. `falsetrue`
C. `truefalse` D. `truetrue`
E. There is no output due to a compile error.

```
String s = "Tortuga";
out.print(s.contains("tor"));
out.println(s.contains("tug"));
```

QUESTION 5

What is output by the code to the right?

A. `true` B. `false`
C. There is no output due to a compile error.
D. There is no output due to a runtime error.

```
boolean p = false;
boolean q = false;
out.println(!(p^q));
```

QUESTION 6

What is output by the code to the right?

A. `14.0` B. `14`
C. `15.0` D. `15`
E. There is no output due to a compile error.

```
out.printf("%.1f",Math.sqrt(225));
```

QUESTION 7

What is output by the code to the right?

A. `0 88 3.0` B. `0 120 3.0`
C. `2 86 3.14` D. `2 118 3.14`
E. There is no output due to an error.

```
int x = 15;
int y = 'X';
double z = 3.14;
y -= x %= z;
out.println(x+" "+y+" "+z);
```

QUESTION 8

What is output by the code to the right if the values for <input1> and <input2> were "xoxoxo" and 2?

A. 2 B. 3 C. 4

D. 5 E. 7

```
String s = <input1>;
int k = <input2>;
int sum = 0;
switch(s.substring(k))
{
  case "xoxo"  : sum+=4;break;
  case "oxoxo" : sum+=3;
  case "xo"    : sum+=2;break;
  case "x"     : sum+=1;
  case "o"     : sum *= 10;
}
out.println(sum);
```

QUESTION 9

In the code to the right, what values for <input1> and <input2> would result in an output of 10?

A. "o" 0 B. "xox" 2

C. "xoxo" 0 D. None of these

E. More than one of these.

QUESTION 10

What is output by the code to the right?

A. 0 B. 1 C. 6

D. 7 E. 8

```
int j = 10000000,c=0;
do{
   j/=10; c++;
  }while(j>1);
out.println(c);
```

QUESTION 11

What is output by the code to the right?

A. 6.6 B. 8.0 C. 8.8

D. 13.2 E. 19.8

```
  double [] list = {1.1,2.2,3.3};
  list[1]=list[2]*2;
  list[2]=list[1]*3;
  out.printf("%.1f\n",list[2]);
```

QUESTION 12

Consider the data file below and code segment to the right. Assume the Scanner f object has been correctly linked to the file shown below. What is the last output of the code to the right?

```
3
The Cosmos is all that is
or ever was, or
ever will be.
```

A. The B. or

C. ever D. ever will be

E. or ever was, or

```
Scanner f = <link to data file>;
out.println(f.nextInt());
out.println(f.nextLine());
out.println(f.nextLine());
out.println(f.next());
```

QUESTION 13

What is output by the code to the right?

A. 0 B. 1

C. 2 D. 3

E. 4

```
double x = 0.0; int y = 0;
double z = Math.toRadians(360);
do{
     x+=Math.PI;
     y++;
}while(x<=z);
out.println(y);
```

QUESTION 14

What is output by the code to the right?

A. true B. false

C. There is no output due to a compile error.

D. There is no output due to a runtime error.

```
boolean b = true && false | true;
out.println(b);
```

QUESTION 15

What is output by the code to the right?

A. 4 B. 8 C. 16

D. 32 E. 64

```
out.println(Double.SIZE);
```

QUESTION 16

What is output by the code to the right?

A. TomDickHarry

B. TomDickLarry

C. DickHarryHarry

D. DickHarryLarry

E. TomMoeLarry

```
ArrayList<String> list = new
 ArrayList<String>();
list.add("Tom");
list.add("Dick");
list.add("Harry");
list.add("Larry");
list.add("Moe");
list.add("Curly");
out.print(list.get(1));
Collections.sort(list);
out.print(list.get(2));
Collections.reverse(list);
out.println(list.get(3));
```

QUESTION 17

Question Omitted

QUESTION 18

Which of the following is NOT an output of the code segment to the right?

A. 0000 B. 0100 C. 0110

D. 1100 E. 1110

```
for(int p = 0;p <= 1; p++)
 for(int q = 0;q <= 1; q++)
  for(int r = 0;r <= 1; r++)
  {
      boolean P = p==1;
      boolean Q = q==1;
      boolean R = r==1;
      boolean S = (P|Q)&(P&!R);
      int s = S?1:0;
      out.print(""+p+q+r+s+" ");
  }
```

QUESTION 19

What is output by the code to the right?

A. 42 B. 126 C. 976

D. 1976 E. 2014

```
int x = <the year of this UIL test>;
int y = <# of pounds in one ton>;
int z = <square ft in a square yd>;
out.println(x%y*z);
```

QUESTION 20

What is output by the code to the right?

A. 67 B. 604 C. 1611

D. 3021 E. 4028

```
int y = 2014;
out.println(y<<3>>>2<<4/10);
```

QUESTION 21

What is output by the `client code` to the right?

A. `160.00` B. `400.00` C. `520.00`

D. `560.00` E. `590.00`

```
static double myst(double a,double b)
{
  double c = 0;
  if(a>48.0)
     {c+=(a-48)*2*b; a=48;}
  if(a>40.0)
     {c+=(a-40)*3/2*b; a=40;}
  c+=a*b;
  return c;
}
//client code
out.printf("%.2f\n",myst(50,10));
```

QUESTION 22

What is output by the code to the right?

A. `Iwttowns!` B. `IIwttowns!`

C. `Iwttownse` D. `IIwttownse`

E. There is no output due to an error.

```
String s = "I want to win state!";
String [] ss = s.split(" ");
String w = "";
for(String b:ss)
{
  char [] list = b.toCharArray();
  w+=""+list[0]+list[list.length-1];
}
out.println(w);
```

QUESTION 23

What is output by the code to the right?

A. `truetruetrue`

B. `truefalsetrue`

C. `truetruefalse`

D. `falsefalsetrue`

E. `falsefalsefalse`

```
String s = "1a2b3c4d5e";
boolean p = s.matches(".*\\d\\w.+");
boolean q = s.matches(".\\D\\S.*");
boolean r = s.matches("[abc]+");
  out.println(""+p+q+r);
```

QUESTION 24

What is output by `statement 1` in the client code to the right?

A. `3`

B. `4`

C. `5`

D. `6`

E. `7`

QUESTION 25

What is output by `statement 2` in the client code to the right?

A. `5`

B. `6`

C. `7`

D. `8`

E. `9`

```
static int A(int m, int n)
{
  if(m==0)
     return n+1;
  if(m!=0&&n==0)
     return A(m-1,1);
  if(m!=0&&n!=0)
     return A(m-1,A(m,n-1));
  return 0;
}
//statement 1
out.println(A(1,3));
//statement 2
out.println(A(2,3));
```

QUESTION 26

Which of the following concepts is NOT represented by the code to the right?

A. inheritance
B. polymorphism
C. overloading
D. overriding
E. All are represented

QUESTION 27

Which of these best replaces **<statement 1>** in the code to the right?

A. `Comparable o`
B. `Object o`
C. `Ork o`
D. `Mork o`

QUESTION 28

What is output by **`segment one`** in the client code to the right?

A. `Ork 0 3 Mork -1 4 Mork -1 0`
B. `Ork 0 3 Mork -1 4 Ork -1 0`
C. `Ork 0 3 Mork -1 4 Mork -1 4`
D. `Ork 0 3 Mork -1 4 Ork -1 4`
E. There is no output due to an error.

QUESTION 29

What is output by **`segment two`** in the client code to the right?

A. `0 0 0`
B. `1 1 1`
C. `-1 0 -1`
D. `-1 0 1`
E. `1 0 -1`

```
public class Ork implements
     Comparable<Ork>{
  int snark,shazbat,nanu;
public Ork(){}
public Ork(int n, int s, int u){
  snark=n;shazbat=s;nanu=u;
}
public String toString(){
  return "Ork "+(snark+shazbat-nanu);
  }
public int compareTo(<statement 1>){
  int x = snark+shazbat-nanu;
  int y = o.snark+o.shazbat-o.nanu;
  return x>y?1:x<y?-1:0;
}
}
class Mork extends Ork
{
  int nanu;
public Mork(){}
public Mork(int n, int s,
           int u, int a)
{
  snark=n;shazbat=s;nanu=u;
  this.nanu=a;
}
public String toString()
{
  return "Mork "+(snark+shazbat-
nanu);
}
}
//client code
  Ork one = new Ork(1,2,3);
  Mork two = new Mork(1,2,3,4);
  Ork trey = new Mork(1,2,3,4);
//segment one
  out.print(one+" "+one.nanu+" ");
  out.print(two+" "+two.nanu+" ");
  out.println(trey+" "+trey.nanu);
//segment two
  out.print(one.compareTo(two)+" ");
  out.print(trey.compareTo(two)+" ");
  out.println(two.compareTo(one));
```

QUESTION 30

What is output by the code to the right?

A. `000000000000000` (15 zeroes)
B. `111111111111111` (15 ones)
C. `00000000000000000111111111111111` (17 0s, 15 1s)
D. `11111111111111110000000000000000` (17 1s, 15 0s)
E. There is no output due to an error.

```
short s = Short.MAX_VALUE;
String t = Integer.toBinaryString(s);
out.println(t);
```

QUESTION 31

Which of the following correctly replaces `<value>` in the code to the right in order to output the value 1?

A. `3` B. `8` C. `9`

D. `10` E. `1000`

```
int x = <value>;
out.println(1000>>>x);
```

QUESTION 32

Which of the following represents the missing lines **<?>** in the output shown in the code to the right?

A.

```
0x1.cp1
0x1.0p2
0x1.4p2
```

B.

```
0x1.0p2
0x1.4p2
0x1.8p2
```

C.

```
0x1.fp1
0x1.5p2
0x1.9p2
```

D.

```
0x1.0p2
0x1.2p2
0x1.4p2
```

E.

```
0x1.10p2
0x1.12p2
0x1.14p2
```

```
double d = 1.0;
while(d<11.0)
out.println(Double.toHexString(d++));

//partial output
0x1.0p0
0x1.0p1
0x1.8p1
<?>
<?>
<?>
0x1.cp2
0x1.0p3
0x1.2p3
0x1.4p3
```

QUESTION 33

What is output by the code to the right?

A. `18` B. `19` C. `21`

D. `24` E. There is no output due to an error.

```
int [] list = new int[10];
Arrays.fill(list,1,10,1);
Arrays.fill(list,2,9,2);
Arrays.fill(list,3,8,3);
int sum=0;
for(int x:list)
 sum+=x;
out.println(sum);
```

QUESTION 34

What is output by the code to the right?

A. `This tess is ss eass.`

B. `This tesst is sso eassy.`

C. `Thisstess issss eass.`

D. `Thiss tesst iss sso eassy.`

E. There is no output due to an error.

```
String s = "This test is so easy.";
String t = s.replaceAll("s\\w","ss");
out.println(t);
```

QUESTION 35

In the chart to the right, representing the most restrictive bound on the runtime of each process in each scenario, where N represents the number of items in list, how many scenarios have a runtime of O(N)?

A. 0
B. 2
C. 6
D. 8
E. 10

Algorithm	Scenarios/Big O Time Complexity		
	Best	Average	Worst
Quicksort	?	?	?
Mergesort	?	?	?
Heapsort	?	?	?
Bubble Sort	?	?	?
Insertion Sort	?	?	?
Selection Sort	?	?	?

QUESTION 36

Using the same chart, how many scenarios have a runtime of O(N^2)?

A. 6
B. 7
C. 8
D. 9
E. 10

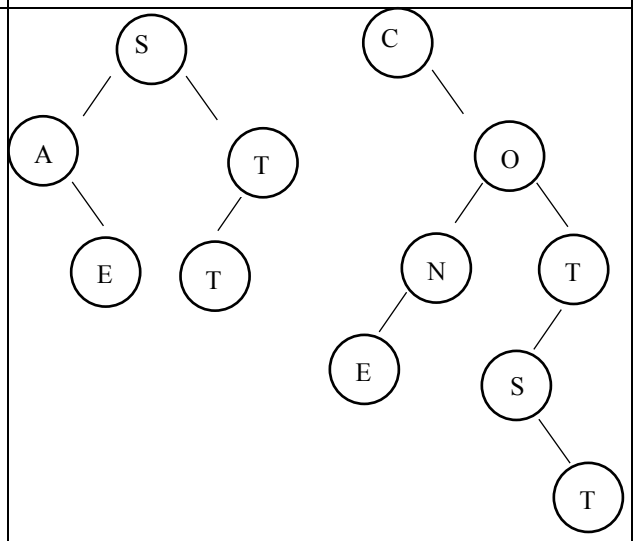
QUESTION 37

To the right is a graph made up of two binary search trees for the strings STATE and CONTEST.

The internal path length of the STATE tree is 6, which means that the total number of steps from each non-root node back to the root is 6. The A and T nodes are each 1 step away, and the E and T nodes are each 2 steps away, for a total of 6 steps.

What is the internal path length of the CONTEST tree ?

A. 6 B. 10 C. 12
D. 15 E. 22



QUESTION 38

How many nodes in this graph (both trees) have only one child?

A. 5 B. 6 C. 7
D. 8 E. 9

QUESTION 39

After the push and pop sequence shown on the right involving two parallel stacks, where the first argument of each command corresponds with the first stack, and the second argument to the second stack, which value would be the next one popped from the second stack?

A. 1 B. 2
C. 3 D. 6
E. 9

```
Push 4 5
Push 1 2
Push 6 3
Pop x y
Push 9 7
Pop x y
Push 5 8
Pop x y
```

QUESTION 40

In a directed graph such as the one on the right, there are often simple paths (no repeated nodes) that form a cycle (back to the starting node), such as these two examples, CGC (also named GCG) and ABCGDA (also named BCGDAB and CGDABC). How many unique cycles are there in this graph?

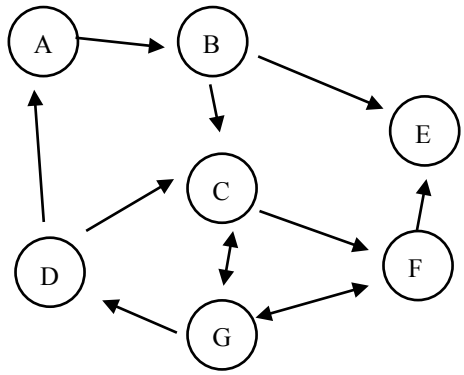
A. 4

B. 5

C. 6

D. 7

E. 8



No Test Material On This Page

Standard Classes and Interfaces — Supplemental Reference

class java.lang.Object
- `boolean equals(Object other)`
- `String toString()`
- `int hashCode()`

interface java.lang.Comparable<T>
- `int compareTo(T other)`
  Return value < 0 if `this` is less than `other`.
  Return value = 0 if `this` is equal to `other`.
  Return value > 0 if `this` is greater than `other`.

class java.lang.Integer implements Comparable<Integer>
- `Integer(int value)`
- `int intValue()`
- `boolean equals(Object obj)`
- `String toString()`
- `int compareTo(Integer anotherInteger)`
- `static int parseInt(String s)`
- `static int parseInt(String s, int radix)`

class java.lang.Double implements Comparable<Double>
- `Double(double value)`
- `double doubleValue()`
- `boolean equals(Object obj)`
- `String toString()`
- `int compareTo(Double anotherDouble)`
- `static double parseDouble(String s)`

class java.lang.String implements Comparable<String>
- `int compareTo(String anotherString)`
- `boolean equals(Object obj)`
- `int length()`
- `String substring(int begin, int end)`
  Returns the substring starting at index `begin` and ending at index `(end - 1)`.
- `String substring(int begin)`
  Returns `substring(from, length())`.
- `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 the specified index.. Returns `-1` if `str` is not found.
- `charAt(int index)`
- `int indexOf(int ch)`
- `int indexOf(int ch, int fromIndex)`
- `String toLowerCase()`
- `String toUpperCase()`
- `String[] split(String regex)`
- `boolean matches(String regex)`

class java.lang.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 java.lang.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, in b)`
- `static int max(int a, int b)`
- `static long round(double a)`
- `static double random()`
  Returns a `double` value with a positive sign, greater than or equal to `0.0` and less than `1.0`.

interface java.util.List<E>
- `boolean add(E e)`
- `int size()`
- `Iterator<E> iterator()`
- `ListIterator<E> listIterator()`
- `E get(int index)`
- `E set(int index, E e)`
  Replaces the element at `index` with the object `e`.
- `void add(int index, E e)`
  Inserts the object `e` at position `index`, sliding elements at position `index` and higher to the right (adds 1 to their indices) and adjusts `size`.
- `E remove(int index)`
  Removes element from position `index`, sliding elements at position `(index + 1)` and higher to the left (subtracts 1 from their indices) and adjusts `size`.

class java.util.ArrayList<E> implements List<E>

class java.util.LinkedList<E> implements List<E>, Queue<E>

Methods in addition to the `List` methods:
- `void addFirst(E e)`
- `void addLast(E e)`
- `E getFirst()`
- `E getLast()`
- `E removeFirst()`
- `E removeLast()`

class java.util.Stack<E>
- boolean isEmpty()
- E peek()
- E pop()
- E push(E item)

interface java.util.Queue<E>
- boolean add(E e)
- boolean isEmpty()
- E peek()
- E remove()

class java.util.PriorityQueue<E>
- boolean add(E e)
- boolean isEmpty()
- E peek()
- E remove()

interface java.util.Set<E>
- boolean add(E e)
- boolean contains(Object obj)
- boolean remove(Object obj)
- int size()
- Iterator<E> iterator()
- boolean addAll(Collection<? extends E> c)
- boolean removeAll(Collection<?> c)
- boolean retainAll(Collection<?> c)

class java.util.HashSet<E> implements Set<E>

class java.util.TreeSet<E> implements Set<E>

interface java.util.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()

class java.util.HashMap<K,V> implements Map<K,V>

class java.util.TreeMap<K,V> implements Map<K,V>

interface java.util.Map.Entry<K,V>
- K getKey()
- V getValue()
- V setValue(V value)

interface java.util.Iterator<E>
- boolean hasNext()
- E next()
- void remove()

interface java.util.ListIterator<E> extends java.util.Iterator<E>

Methods in addition to the `Iterator` methods:
- void add(E e)
- void set(E e)

class java.lang.Exception
- Exception()
- Exception(String message)

class java.util.Scanner
- Scanner(InputStream source)
- boolean hasNext()
- boolean hasNextInt()
- boolean hasNextDouble()
- String next()
- int nextInt()
- double nextDouble()
- String nextLine()
- Scanner useDelimiter(String pattern)

Computer Science Answer Key
UIL Invitational A 2014

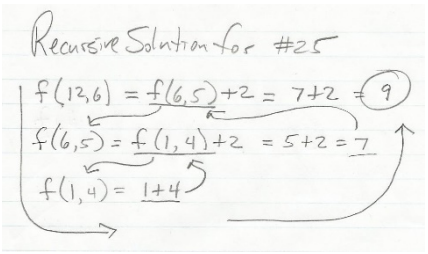
1) D	11) B	21) D	31) C
2) B	12) C	22) A	32) B
3) D	13) D	23) E	33) A
4) A	14) E	24) C	34) D
5) E	15) C	25) D	35) B
6) C	16) A	26) B	36) A
7) B	17) A	27) D	37) B
8) B	18) D	28) B	38) C
9) B	19) B	29) D	39) C
10) C	20) E	30) A	40) 4 last value popped 7 next to be popped

Note to Graders:

- All provided code segments are intended to be syntactically correct, unless otherwise stated (e.g. error is an answer). **Ignore any typographical errors.**
- Any necessary Standard Java 2 Packages are assumed to have been imported as needed.
- Assume any undefined (undeclared) variables have been defined as used.

Brief Explanations:

1. $11110_2 + 11011_2 = 30_{10} + 27_{10} = 57_{10} = 71_8 = 39_{16} = 111001_2$
2. b = 19 + 13 = 32, c = 13
3. Integer object array cannot be initialized with a double
4. j starts at 5, outputs from 4 down to 1, and stops at 1
5. character at position 4 is k
6. list2 is an alias for list 1, so any changes made by one are changes made to the other
7. p^q is p xor q, which requires opposites in order to be true. Since both are true, the result for p is false.
8. This string switch statement matches at "sweet", and outputs both "yum" and "yom" since the break is only after "yom"
9. The maximum of 5.2 and 3.1 is 5.2
10. The lengths of each row of this uneven grid are 3, 4, and 2
11. getNumStrings is an accessor method with a heading of public int since it returns an integer
12. and requires no parameter, so has empty ( ) with NO semicolon!
13. and simply returns the numStrings instance field value
14. Since & evaluates first in bitwise order, 15&7 results in 7, and then 25 | 7 results in 31.
15. This is a loop that calculates the log base 2 of 100…e increments by 1, but f doubles each time and passes 100 at the 7th iteration
16. The parameter in a method call is called the actual parameter
17. Since this is a chain if else, only one value is output for each call, according to the logic of the if statements. 6 produces 30, 9 produces 1, and 7 produces 7.
18. Both the (8) and (8,13) substring calls produce the word "Probe" from this string since the P is at position 8 and 13 is the length of the string, one step PAST the end of the substring desired.
19. This expression follows the order of operations and integer divides 12 and 5 to get 2, then adds 2.5 to get 4.5
20. The Boolean expression is p or q and q, which when simplified just becomes p (Law of Absorption) and therefore each output digit matches the p digit of the term.
21. Both I and II options correctly (but in different ways) isolate the 6. Option III isolates the 9.
22. 2PI is just a full circle..360 degrees.
23. Any integer left shifted 32 spots (the bit size of the integer data type), will simply return to its original value. Essentially it is a Left Circle back to the original number.
24. The list is indeed empty to start with, and after all the action, 3.1 is the sole surviving element in this list at position 0.
25. See the recursive trace on the right for the solution to this problem.



26. The "`[elt]+`" pattern splits at any sequence of the letters e, l and t, which produces the array `["Fr","Fa", "inTomP","y"]` in this instance, producing "`FryFa`" for this output.
27. This ternary operation results in false, since 100%3 is not 0, therefore the resulting string is the one following the :, which is "`bad`", **very "`bad`"**!!!
28. The first different characters in these two strings are 'a' and 'i', which produces -8 since 'a' is 8 places before 'i'.
29. Since the value 10 can only map to one value, the "ten" is replaced with "sepuluh", another word for 10. Look it up!
30. Another digital electronics question! Don't you just love 'em? Just learn the basic symbols and this will become very easy for you. The bullet shape is the AND, and the arrow is OR. This is simply A and B or C.
31. DeMorgan's law is applied to the **not(a or b)**, resulting in **not a and not b**, which when "anded" with another **not a** simply becomes **not a and not b**.
32. Binary search is easy. Find the middle, and if doesn't match, go left or right and find the middle again. Repeat this process until the middle is the one you want. Then count the "middles" and that's how many steps you took to find it.
33. This structure is most certainly valid. Any class inheriting an abstract class is required to implement any abstract method in that class, so class B is REQUIRED to implement method one() from class A. Anything else is optional.
34. This is just simply 4 from method one times 5 from method two times the 2 from the variable x, for a result of 40.
35. All of these implementations of class B are valid. Look them over carefully.
36. Since p.next pointer references the second node of the list, and the data for that node is 2, the resulting output is 2.
37. In this TreeSet process, 4 is added twice, but since there are no duplicates in sets, only remains once. The 6 is removed, leaving only the 4, 5, and 7, so the size is 3 and 6 is not in the list.
38. Data input is a classic use of the try catch block. Since 3.14 is a mismatch for integers, the exception is thrown by the try block and caught by the catch block, resulting in the "Bad data. " output. The finally block ALWAYS occurs, no matter what.
39. In the heapify process of a min heap, the process always starts at the bottom right of the tree, working left and upwards, switching any parent and child values that are not in correct min heap order. The first such occurrence here is the 6 and 3. Next will be the 1 and 2, and so on.
40. In this queue push and pop sequence, the 3,5, and 4 are pushed, then the 3 is popped, push the 7, pop the 5, pop the 4, then push the 9. The 4 was the last value popped, and the 7 sits at the front of the queue, waiting to be popped next.

Computer Science Answer Key
UIL Invitational B 2014

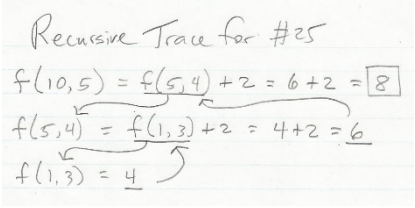
1) B	11) A	21) E	31) B
2) A	12) B	22) D	32) A
3) D	13) B	23) E	33) A
4) C	14) A	24) B	34) C
5) C	15) E	25) D	35) B
6) C	16) D	26) B	36) E
7) B	17) D	27) E	37) C
8) C	18) D	28) D	38) B
9) C	19) E	29) E	39) A
10) A	20) B	30) D	40) 5 last value popped 8 next to be popped

Note to Graders:

- All provided code segments are intended to be syntactically correct, unless otherwise stated (e.g. error is an answer). **Ignore any typographical errors.**
- Any necessary Standard Java 2 Packages are assumed to have been imported as needed.
- Assume any undefined (undeclared) variables have been defined as used.

Brief Explanations:

1. $110_2 + 100010_2 = 6_{10} + 34_{10} = 40_{10} = 50_8 = 28_{16} = 101000_2$
2. H = 24/5 = 4 (integer division)
3. Even though this is an array of Double objects, autoboxing **does not apply** when instantiating a static array like this. The 4 is an int and will cause a compile error, "incompatible types"
4. k starts at 3, outputs 6, 9 and 12, and stops at 12
5. the character 98 is the letter 'b', at position 5 from position 1. The 'b' in position 0 is not considered.
6. By the end of this assignment sequence, 4.5 is the element in every position.
7. p^q is p xor q, which requires opposites in order to be true, therefore p=true;p=false or p=false;q=true; will both evaluate to true.
8. The resulting values for all five choices are: "a",-1,"aa",0,"bb",3,"cccc",-1,"",-1
9. The minimum of -5.2 and 3.1 is -5.2
10. The 6 is in the second row (row 1), and in the third position of that row (column 2).
11. **`setNumStrings`** is a mutator method with a heading of **`public void.`**
12. It receives an integer **`(int n)`**
13. and assigns it to numStrings (**`numStrings = n;`**)
14. Since shift operations have priority over bitwise operations, 15 is left-shifted first, becoming 30, then 30 xor 30 is zero (Any value xor itself is zero – in assembly language that is one way of assigning a value of zero to a register).
15. The j values in this loop sequence are: 0, 0, 1, 3, 4, 12, 13, 39, 40, 120 and finally 121.
16. As is evident in the heading, this method is both a static method and a return method.
17. Since this is a chain if else, only one value is output for each call, according to the logic of the if statements. 9 produces 5, 8 produces 3, and 14 produces 5.
18. The (6,7) substring call is the correct one to access the letter "R".
19. This expression follows the order of operations, where 60%9 produces 6, and then is subtracted from 31 to make 25.
20. The Boolean expression is **p and q or q**, which when simplified just becomes **q** (Law of Absorption) and therefore each output digit matches the **q** digit of the term.
21. 28.5 mod 9 produces the value 1.5.
22. 180 degrees in radians is PI.
23. Any integer left shifted 32 spots (the bit size of the integer data type), will simply return to its original value. Essentially it is a Left Circle back to the original number. The Integer.toBinaryString method only outputs significant digits…no leading zeroes.
24. Since this the type of this ArrayList is not designated, any mixture of objects, including null, is acceptable. The final contents of the array are [null, 6, "ball", and 4.7].
25. See the recursive trace on the right for the solution to this problem.
26. The "`[pote]`" pattern splits at any of the letters 'p', 'o', 't' and 'e', which produces the array `["il","v","","","","ain"]` in this instance, producing "`vain`" for this output.
27. This ternary operation results in true, since 100%5 is equal to 0, therefore the resulting string is the one following the ?, which is "`walking`".
28. The first different characters in these two strings are 'r' and 'c', which produces 15 since 'r' is 15 places after 'c'.



29. Since hash structures guarantee no certain order, there is no indexing, therefore the call get(0) does NOT return the first element of the hash mapping, but simply looks for the mapping of the key value zero, and finding none, returns null.
30. Since the OR happens before the AND in this case, and AND occurs before OR in logic order, it necessary to use parentheses, producing (A OR B) AND C.
31. A ⊕ B simplifies to (not A and B or A and not B), which when FOILED with (A+B) produces the same thing, A ⊕ B.
32. The first "middle" found is the 6. Since 7 is to the right, the next "middle" is 9, then going left where the final "middle" is 7, the search item.
33. This structure is most certainly valid. Any class inheriting an abstract class is required to implement any abstract method in that class, so class B is REQUIRED to implement method two() from class A. Anything else is optional.
34. This is the product of 5 from method one, 2 from method two, and 2 from the variable x, for a result of 20.
35. The word "implements" is used when an interface is used, therefore option II is not valid. All of the rest are valid.
36. Since p.next.next pointer references the third node of the list, and the data for that node is 9, the resulting output is 9.
37. In this TreeSet process, 4 is added twice, but since there are no duplicates in sets, only instance remains. When the 4 is removed, leaving only the 5, 6, and 7, the size of the set is 3 and 6 is indeed in the list, resulting in the output **3true**.
38. Data input is a classic use of the try catch block. Since 3.14 works for doubles, no exception is thrown, and program flow drops to the finally block, which always executes, no matter what.
39. In the heapify process of a max heap, the process always starts at the bottom right of the tree, working left and upwards, switching any parent and child values that are not in correct max heap order. The first such occurrence here is the 6 and 7. Next will be the 2 and 9, and so on.
40. In this queue push and pop sequence, the 9 and 7 are pushed, then the 9 is popped, push the 5, 8, and 6, pop the 7 and the 5. The 5 was the last value popped, and the 8 sits at the front of the queue, waiting to be popped next.

Computer Science Answer Key
UIL District 1 2014

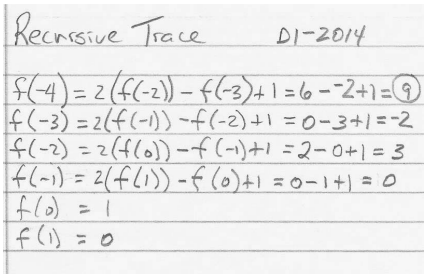
1)	A	11)	B	21)	E	31)	E
2)	A	12)	B	22)	D	32)	B
3)	D	13)	C	23)	D	33)	A
4)	D	14)	B	24)	A	34)	E
5)	B	15)	D	25)	E	35)	B
6)	D	16)	C	26)	B	36)	B
7)	E	17)	D	27)	D	37)	D
8)	D	18)	E	28)	A	38)	D
9)	B	19)	C	29)	A	39)	C
10)	A	20)	C	30)	C	40)	A

Note to Graders:

- All provided code segments are intended to be syntactically correct, unless otherwise stated (e.g. error is an answer). **Ignore any typographical errors**.
- Any necessary Standard Java 2 Packages are assumed to have been imported as needed.
- Assume any undefined (undeclared) variables have been defined as used.

Brief Explanations:

1. $10101_2 + 10000_2 = 21_{10} + 16_{10} = 37_{10} = 45_8 = 25_{16} = 100101_2$
2. For Boolean AND to be true, both inputs need to be true
3. The Math.ceil method returns the "rounded up" decimal value, in this case, 4.1573 goes up to 5.0
4. 13.7 times 2 is 27.4
5. The 'm' is replaced with 'k', making the new String "bikinitop"
6. 9/2 is 4, 6.5 * 2 is 13.0, and 4 – 13.0 is 9.0.
7. This is a runtime error (null pointer exception) since null cannot be added to an integer.
8. ~ means complement, or simply put, opposite, minus 1. ~50 is -51. -51/7 is -7, which is then multiplied by 8 (<<3) making -56. ~(-56) becomes 55.
9. The matching case for 'e' outputs the word "Dude", and stops at the break
10. This code counts all the letters in "red", "white", and "blue" that are NOT in "yellow", which are "rdhitbu"
11. The job of the toString method is to include all of the instance values of the object in some form, so the return statement that does that is the best answer, even though **return "6 string acoustic"** will do the job for this particular object, but not for an object with different values.
12. The Object class is the origin of the toString method
13. Overriding is the process of redefining a method inherited from a parent class. Overloading is when you have several methods in the same class with the same name, but different parameter signatures.
14. This method simply adds up the digits in each number. 637 and 790 both have the greatest sum, but since 637 came first, it is the answer.
15. This loop never happens since x==0 evaluates to false at the beginning, so there is no output.
16. List position 2 gets the value 6 (the element in position 5), and list position 1 gets the value 5 (the element in position 4).
17. This method simply calculates and returns the 3rd side of a right triangle...
18. …which is the Pythagorean theorem.
19. The first ***different*** letters in these two strings are 'u' and 'l', and 'u' has an ASCII value 9 greater than 'l'.
20. The hex value B4 is simply 11(B) times 16, or 176, plus 4, or 180, which has a binary value of 10110100.
21. Since 24 mod 7 is 3, the ternary operator evaluates to false, and 24+3 is the result.
22. Any integer right shifted 32 positions is back to where it started, actually a right circle 32 to be precise. The binary value of 100 is 1100100.
23. The log of E is 1.0.
24. The minimum value for an int is -2147483648, which in binary is 1 with 31 zeros.
25. See the recursive trace on the right for the solution to this problem.
26. List1 only adds even numbers, while List2 adds all of them. The removeAll indeed removes all of the evens from List2, leaving the odds, but the output only asks for List1, which contains the evens.
27. This sequence effectively pushes three characters in priority order, then pops the front two, and repeats this process throughout the end of the string. In the first three, "UIL", the "U" remains since "I" and "L" are alphabetically in front of the "U", so they get popped.



28. Even though j is an int, the /= shortcut has an automatic cast, so 100 divided by 20.0 still returns 5. k gets 20.0 / 5, which is 4.0.
29. The Boolean expression P OR Q XOR P simplifies to just P OR Q, which results in true for all combinations except for false false.
30. This is a simple Digital Electronics diagram, with A and NOT B going into a NOT AND gate, so the expression is NOT(A AND NOT B).
31. An interface requires all methods to be designated public for it to compile, so the fix is to put the word "public" before each method in both the interface and the class. {} is a sufficient implementation for the A1 void method, which is the way you would simply ignore a method you do not wish to implement with anything significant.
32. Given the description of what each method should do, the output here is obvious…"HelloWorld0".
33. A TreeMap is similar to a mathematical function, in the fact that there can only be one mapping per key (for every x this is one and only one y). There can be, however, duplicate values, like the 7 mapped by both "b" and "f". When the "c" is mapped again with the 3, the 4 is removed. But then the "c" mapping is removed altogether, so there is no "c" mapping at the output. Since it is a Tree mapping, the keys are in natural order.
34. The most efficient of all Big O classifications is O(1).
35. The sequence is this: push 3, push 5, push 9, pop 9, push 6, pop 6, pop 5, push 2, and push 7. The 5 was the last value popped.
36. A*0 is simply false, and goes away. B OR 1 simplifies to true since OR with true is always true, therefore the simplified expression here is just TRUE, or 1.
37. The diagonal spanning from row 7, col 1 up and to the left to row 1, col 7 has 7 1s in it, the longest in this matrix.
38. a%10 results in 5, b/10 is 3, and b%10 is 4. The sum 5+3+4 is 12.
39. An adjacency matrix is a classic way to express a graph situation. Study the example carefully and it will make sense.
40. The sequence of values through the loop execution are: 5.0 and 20.0 to start, then 12.0 and 18.0, 14.0 and 17.0, 16.0 and 16.0, and finally 16.0 15.0.

Computer Science Answer Key
UIL District 2 2014

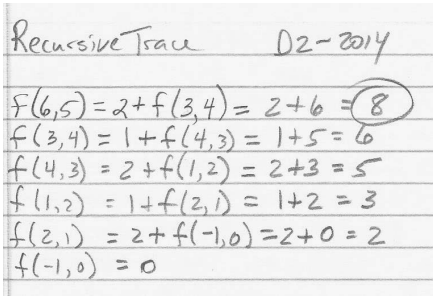
1) B	11) B	21) C	31) A
2) A	12) A	22) A	32) C
3) E	13) D	23) E	33) E
4) D	14) B	24) C	34) B
5) D	15) D	25) A	35) C
6) A	16) E	26) A	36) E
7) D	17) B	27) C	37) D
8) E	18) D	28) A	38) D
9) E	19) C	29) B	39) B
10) B	20) A	30) B	40) C

Note to Graders:

- All provided code segments are intended to be syntactically correct, unless otherwise stated (e.g. error is an answer). **Ignore any typographical errors**.
- Any necessary Standard Java 2 Packages are assumed to have been imported as needed.
- Assume any undefined (undeclared) variables have been defined as used.

Explanations:

1. $100010_2 + 100000_2 = 34_{10} + 32_{10} = 66_{10} = 102_8 = 42_{16} = 100010_2$
2. This is simple arithmetic. Just remember the data types for the output.
3. The first true result is obvious since both x and y reference the same object. For the **`y=5`** reassignment, there is a common memory section in Java for Strings and for smaller value integers that objects share when they are instantiated simply with the equals sign. Therefore, even though it looks like a separate object is created, it simply references the 5 that is in common memory, and therefore it is still pointing to the same memory location. However, when the **`new`** operator is used, a separate memory location is used, which results in **`false`** for the == operator.
4. Since the ++ is a post-increment operator, the value is output first, then the variable is incremented, with the result shown.
5. The **`lastIndexOf`** method is straight forward…the last index of the letter 'a' is in position 8 of the string.
6. Remembering that Java lists use zero based indexing (first element is in position zero), the elements in position 1 and 3 are 3 and 2, whose sum is 5.
7. The only way for the OR (||) operator to be false is when both Boolean values are false.
8. Both output statements are executed here, the first one because the ***`if`*** statement is true, and the second one regardless of the if statement since it is not attached to it, despite the indentation. The resulting output is simple math.
9. Right shift 2 is essentially dividing by 4 (2^2), and left shift 2 is multiplying by 4, with obvious results.
10. The floor function returns the nearest lower whole number value of the decimal, in this case, -6.0.
11. The traditional modifier method of classes starts with the word **`set`**, and in this case **`setNumStrings`** is the method to use, giving it the desired value as a parameter.
12. Similarly, the word **`get`** is the traditional prefix for accessor methods of instance variables, therefore **`getType`** is the one to use in this situation.
13. The toString method in this class definition lists the type first, followed by a colon, then the number of strings, and the word "string".
14. This is simple arithmetic. Nuff said.
15. This **`showGrid`** method outputs the entire matrix from bottom row to top, in right to left column order.
16. The value 260.0 divides into 130.0, 65.0, 32.5, 16.3, 8.13, 4.1, 2.03, 1.02, and finally 0.51, with 9 divisions. 250 requires only 8 divisions, and 600 requires 10.
17. The contents of the array at the start are: 0 5 2 0 0 0. After each loop iteration the contents are: 0 5 2 3 0 0, 0 5 2 3 -1 0, and 0 5 2 3 -1 4. Position 4 contains -1 at the conclusion of the method call.
18. The greatest value at the end is 5.
19. Since the length of the string is 20, the substring calls with 15, 5 and 10, and 7 and 12 all will return a string of length 5.
20. The expression **`p xor q and p`** simplifies to **`p and not q`**, which means the only true result is when p is true and q is false, indicated by 101 in the output. Using Boolean identities, the simplification sequence is as follows: **`p^q&&p = p&&!(q&&p) || !p&&q&&p = p&&(!q||!p) = p&&!q`**. You can also use the truth table process to evaluate this expression.
21. 42.0 % 13 results in 3.0, which is then incremented to become 4.0.
22. Decimal 10 in binary is 1010.
23. The short data type is stored in 16 bits of memory.
24. The natural log of E (**`2.718281828459045`**), the base of the natural logs, is 1.00.
25. The recursive trace for this question is shown on the right.
26. The binary representation for -1 is a string of 32 1s, which when right shifted 32 places circles back to the same 32 1s.
27. The split for this problem results in the following: `[IL, veA, a, ade]`, with a length of 4 and "`ade`" in position 3.
28. The base 5 equivalent of 34 is 114.
29. The replaceAll method does not change the existing String (Strings are immutable), but instead returns a new String with the modifications indicated. The original String w is not changed, however a new String **s** is created changing all 'n's to 'm's.



30. The A and B signals go into a NOR gate, which goes into an XOR gate with C, resulting in NOT(A OR B) XOR C.
31. This code is fine as is. Unlike the interface, the abstract class does not require the word `public` preceding the method name.
32. The call to methods A1 and A2 simply result in the output, "`I made a 240`".
33. The contents of the queue after each command are: [3], [3, 5], [3, 5, 9], [5, 9], [5, 9, 6], [9, 6], [6], [6, 2], [6, 2, 7], with 6 at the front.
34. The least efficient of these O(N) ratings is $O(N^2)$, which is typically characterized by some nested loop process, such as an insertion sort or bubble sort.
35. Although there are 16 words in this sentence, only 14 are unique, which is what this code does (sets have no duplicates).
36. The expression A AND B AND A OR 0 simplifies to just A AND B, since the repeated A dissolves into just one A, and the OR 0 is the identity rule and effectively disappears.
37. Since 97 is the ASCII value for lower case 'a', 100 represents 'd', which is where this diagonal of characters starts, producing the series "defgh".
38. The contents of this list after each command is as follows: [], [4], [4, 5], [4, 5, 6], [4, 5, 6, 5], [4, 5, 6, 5, 7], [5, 6, 5, 7].
39. To find out the number of 1s in this matrix, simply count the number of arrows, which is 6. Since it is a 4X4 matrix, which means 16 elements, the remaining 10 elements are zeroes.
40. This one is tricky. The first two statements in the p method actually effect the actual parameters, the lists x and y, since arrays are passed by reference, but the third statement (a=b) does not. Even though **a** is reassigned to reference the **b** list in the method, this does not make the original x reference change, therefore it still points to its original list. Here is the state of each list after each command.
    - x[0] = 10 y[0] = 5
    - a[0] = 15 b[0] = 5
    - a[0] = 10 b[0] = 10
    - x[0] = 15 y[0] = 10
    - a[0] = 25 b[0] = 15
    - a[0] = 10 b[0] = 10
    - x[0] = 10 y[0] = 25

Computer Science Answer Key
UIL Region 2014

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

Note to Graders:

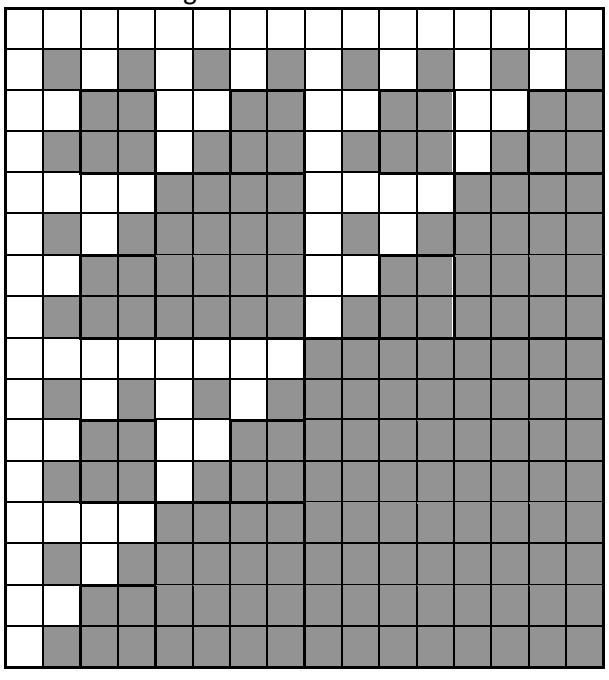
- All provided code segments are intended to be syntactically correct, unless otherwise stated (e.g. error is an answer). **Ignore any typographical errors**.
- Any necessary Standard Java 2 Packages are assumed to have been imported as needed.
- Assume any undefined (undeclared) variables have been defined as used.

This page intentionally left blank

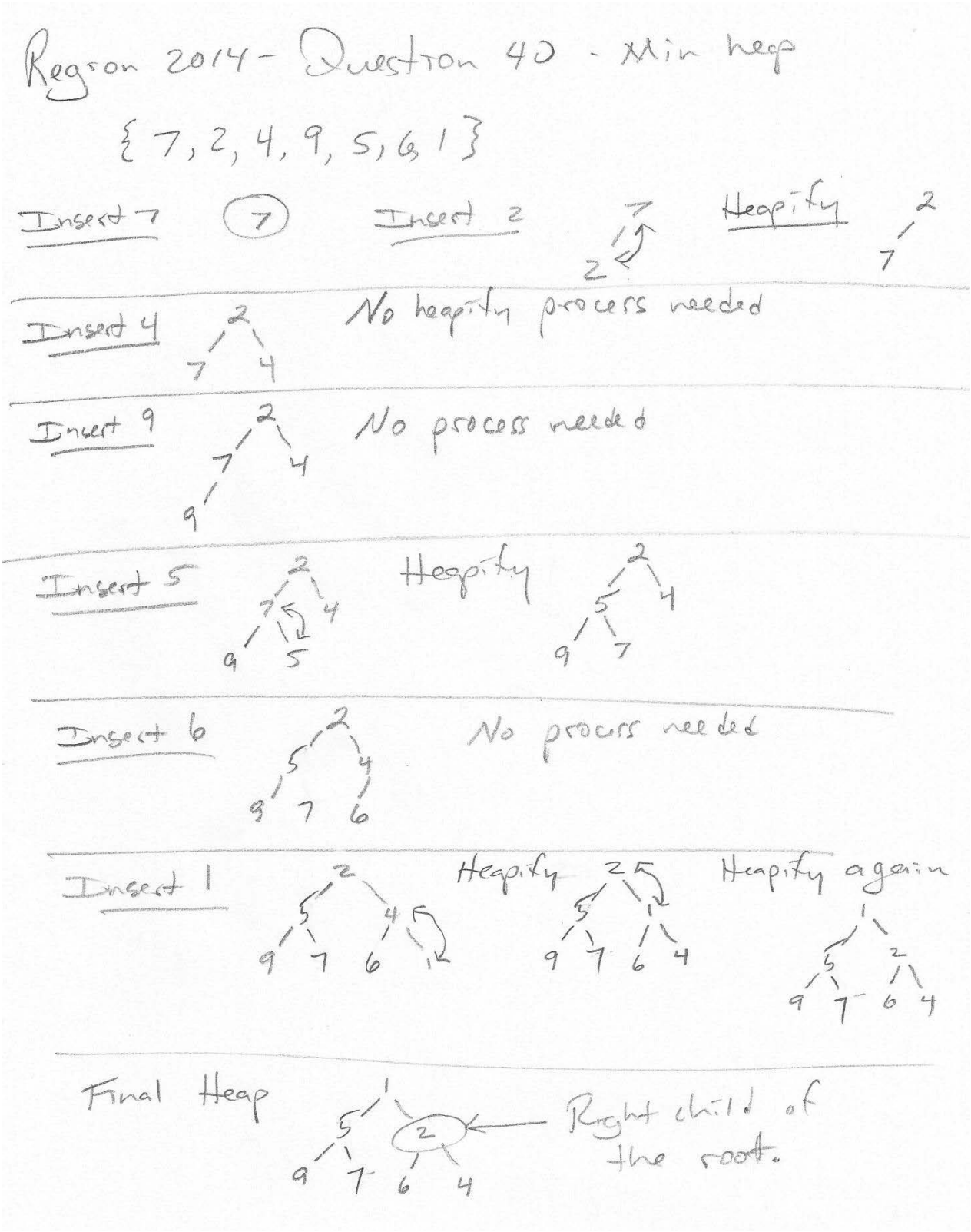
Explanations:

1. $111011110_2 - F2_{16} = 236_{10} = 354_8 = EC_{16} = 11101100_2$
2. Using order of operations, 16%9 goes first (7), then 4*0.2 (0.8), then 7-0.8, which equals 6.2.
3. Since every Java data type has a String representation, the "%s" printf format specifier can take any data type, and this code works, with `HelloGoodbye` on one line, and `4_true` on the next line.
4. This method calls returns a String in all lowercase, but does not change s (since Strings are immutable) and since it was not reassigned to s, the output is the original value of s, `BalloonBomb`.
5. This Boolean expression is equivalent to P or Q, which is false only when both P and Q are both false.
6. Since Math.round returns a long when a double parameter is given, long is the best data type to use, even though a double could take it.
7. 70 divided by 3 equals 23. Add that to 5.2 and you get 28.2.
8. Since 'B' and 'b' are different values in the ASCII character map, they are not equal.
9. When c has a value of 14, the next column (15) is the first time water will be detected.
10. Since the default value of char Arrays is the zero value space (not the 32 value space), the values added here are 65+0+50, for a total of 115.
11. File and Scanner are the two main classes used for input. FileWriter and PrintWriter are used for file output.
12. The a and b values for this loop sequence are: 0 2.4, 4 4.8, 13, 9.6, 32 19.2. The last pair causes the loop to terminate since the sum is not less than 25. Those values are the ones output.
13. This tests your knowledge of operator precedence, something you should study and know very well. Since the + operator precedes the <<, the 2 and 1 are added first, then a <<3 is applied to 5, in essence multiplying 5 by 2^3, or 8, which equals 40. The common error is to shift first, then add 1, which would result in 21.
14. The float data type uses 32 bits of memory.
15. This question is about how the remove method works. Does it remove the value 2, or the value in position 2? It is the latter, the value in position 2, which is the 3. The resulting list contains [4, 1, 2].
16. The best way to remember the Digital Electronic shapes is this: bullet shapes (flat back) are AND, arrow shapes (curved back) are OR, double arrows are XOR, and a small circle means NOT. The expression for this one is A XOR NOT(B AND C) OR NOT D.
17. This double ternary operator works just like a nested if else statement: if(a>50) if(a<75) output "red" else output "green" else output "blue". "red" will be output for 51 through 74 (24 times). "blue" is output for the values 45 through 50, and "green" for all the values 75 and beyond.
18. The first match is false. In the match string, ".[^WIN]+.*", the dot means a single character, which correctly matches the "U". "[^WIN]+" means match one of more characters NOT in "WIN". The "I" causes this to be false, therefore the match is false. The match string with all the dots is an exact match since the length is the same, and a dot matches any character, there it is true.
19. The object method is always called when it overrides the super class method, which in this case outputs, "The dog is a: dachshund". Since it is not possible to determine which show method will be called (Animal reference could be reassigned during the execution), the show method called is determined during run time, an example of late (dynamic) binding.
20. When both base and derived classes have member values of the same name, the base class member value is always called by default. This is an example of early (static) binding.
21. This code demonstrates the difference between early and late binding. Statement C is the only one that is true. Statements A and B are reversed. Early binding occurs at compile time, and question 20 is an example of this. Question 19 is an example of late, or dynamic binding.
22. The somesort method is first called with the values 0 and 5, which gives a middle value of 2. Subsequent calls and middle values produced are: (0,2)->1, (0,1)->0, (3,5)->4, (3,4)->3. The (2,2) and (5,5) calls result in no middle values output.
23. This is the merge sort.
24. The Big O rating for this is O(N log N).
25. The array sizes produced in this matrix are of size 2, 4, 6, 8, and 10, for a total of 30 slots.
26. The decimal value 75 in base 7 is 135.
27. The priority queue step-by-step sequence produce by this code is: ~~R,~~ ~~e~~g, ~~g~~io, ~~space~~ino, ~~I~~Uino, ~~space~~LUino, ~~0~~2LUino, ~~1~~24LUino, resulting in final queue of "24LUino".
28. Since the order of precedence for bitwise operators is AND, XOR, then OR, the 4 AND 7 goes first, resulting in 4. 13 XOR 4 is 9, and then 9 OR 6 is 15.
29. This Boolean expression simplifies to !P&!Q | P&!Q, which means 00 and 10 are the only two ordered pairs that result in true.
30. This expression simplifies to 196 – 12 * 2, which equals 172.
31. To the right is the recursion diagram for this question. →
32. Preorder traversal starts at the root and outputs each element in "touch left side" traversal order.
33. The array contents, step by step, are as follows: [], [4], [4 5], [4,5,1,3,5,6,7], and [4,5,1,3,6,7].
34. A, B, and C are correct as stated.
35. The Boat constructors include the default constructor, Boat(), and three one parameter constructors.
36. The toString method constructs and returns the String designed to output the member values of this boat.
37. Although all three code segments will accomplish the desired output, the most direct Option is II, where the parameter values match the required data types exactly. Option I works OK, but has to make a data type promotion in the call to the setDraft method.Option III works as well, but is just a terribly inefficient way to accomplish it.
38. This method simply adds the ASCII values of the characters in the given string, making negative any values that are odd. "bed" returns the sum of 98 – 101 + 100 = 97.
39. "MET" returns the sum -77 – 69 + 84 = -62.

Recursion Diagram – #31



40. Below is the step by step process for this min heap process.



Computer Science Answer Key
UIL State 2014

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

Note to Graders:

- All provided code segments are intended to be syntactically correct, unless otherwise stated (e.g. error is an answer). **Ignore any typographical errors.**
- Any necessary Standard Java 2 Packages are assumed to have been imported as needed.
- Assume any undefined (undeclared) variables have been defined as used.

This page intentionally left blank

Explanations:

1. $527_8 + 910_{10} = 1253_{10} = 2345_8 = 4E5_{16} = 10011100101_2$
2. 23 / 4 + 9.4 % 3 = 5 + 0.4 = 5.4
3. The %s format specifier of the printf statement accepts the string representation of any data element, and any extra parameters, such as the "true" string in this statement, are ignored.
4. The contains method of the String class returns true if there is an exact match of the given parameter somewhere in the string. "tor" does not match due to the uppercase "T", but "tug" does.
5. This expression, NOT(P XOR Q), simplifies to be P equals Q, which means that for the expression to evaluate to true, P and Q must be the same value, either both false or both true.
6. The square root of 225 is 15, and since the Math.sqrt method returns a double, and the %.1f format specifier is used, the output is 15.0.
7. A compound expression evaluates from right to left, therefore x changes from 15 to 2 (15 % 3.14 = 2.44, which is autocast to 2), y gets the ASCII value of 'X' (88), which is then subtracted by 2, resulting in 86. The value of z does not change.
8. The String produced by `"xoxoxo".substring(2)` is `"xoxo"`, which matches the first case in the switch, resulting in a value of 4 for sum.
9. The only input values that produce the value 10 are "xox" and 2, resulting in the String "x", which matches the fourth case, adds 1 to sum, then drops down to the next case and multiplies sum by 10, resulting in an output value of 10.
10. This is essentially the calculation of the log, base 10, of 10,000,000, which is 7. This means that 10,000,000 is equal to 1 X 10^7.
11. The sequence of calculations is as follows:
    - list[1] (currently 2.2) is assigned the value 3.3*2, which is 6.6
    - list[2] (currently 3.3) is assigned the value 6.6*3, which is 19.8
12. The input sequence is as follows:
    - `nextInt` grabs the 3
    - `nextLine` gobbles up the whitespace after the 3
    - `nextLine` takes the String, `"The Cosmos is all that is"`
    - `next` gets the String, `"or"`, the last output
13. Since `Math.toRadians(360)` produces the value 2PI, it will take the variable x three additions of PI to exceed that value, therefore the value of y after the loop is 3.
14. Even though this expression is using the `&&` logic operator and the `|` bitwise operator (which really evaluates first in the Java order of precedence), the result is the same…true. False OR True is True, and then True AND True is True.
15. The double data type uses 64 bits of storage.
16. After all of the elements are added, "Dick" is in position 1. Once the list is sorted, "Harry" is in position 2, and after the list is reversed, "Harry" is in position 3.
17. Omitted
18. The output of 1100 should be 1101, which shows that when P = true, Q = true, and R = false, (P OR Q) AND (P AND NOT R) evaluates to `(true OR true)AND(true AND NOT false)`, which simplifies to `true AND (true AND true)` which further simplifies to `true`, not `false` as 1100 shows.
19. With the values 2014 (year of this test), 2000 (pounds in one ton), and 9(square feet in one square yard), the expression is 2014%2000*9, which evaluates to 14*9, or 126. Mod has the same level of precedence as times, therefore it occurs first and is evaluated first.
20. Since division has a higher precedence than shift operations, 4/10 evaluates to 0, which causes y to have a net left shift of 1, which means 2014 is multiplied by 2^1, resulting in 4028. The most likely error of 1611 would be the result if the shift operations went first (net left shift 8), with the shift result of 16112 integer divided by 10 to make 1611.
21. This method calculates a person's paycheck, with double overtime for any hours over 48, time-and-a-half for hours over 40 up to 48, and regular pay for 40 hours or less. For 50 hours at $10 an hour, the pay is calculated at 2 hours at double time (2*2*10), plus 8 hours at time-and-a-half (8*1.5*10), plus 40 hours at regular pay (40*10), for a total paycheck of $560.00.
22. After the initial " " split into the array of Strings, the for loop takes the first and last character of each word and builds a new String. `"I want to win state!"` produces `"IIwttowns!"`. The "I" counts twice since it is both the first letter and last letter of that word.
23. The first match `(".*\\d\\w.+")` looks for zero or more characters + a single digit + single word character + one or more of any character, which is `true` since the "**.***" can be ignored, the "1" matches the single digit, the "a" matches the single word character, and the rest of the string matches the "**.+**". The next match `(".\\D\\S.*")` means a single character + single non-digit + single non-space + zero or more characters, also `true`. The third match (`"[abc]+"`) checks for one or more characters from the `[abc]` set, and nothing more. This match is false since the first character is NOT from the `[abc]` set.
24. In computability theory, the **Ackermann function**, named after Wilhelm Ackermann, is one of the simplest and earliest-discovered examples of a total computable function that is not primitive recursive. All primitive recursive functions are total and computable, but the Ackermann function illustrates that not all total computable functions are primitive recursive. See the recursive trace below. As you can see, it doesn't take much for this recursive process to get out of hand. As an example, A(4,1) will most likely cause a stack overflow if you try to run it on a PC, and just forget about even trying to trace it by hand!
25. Same as 24.
26. All of these are classic Object Oriented Programming concepts. Inheritance is represented by the Mork class extending the Ork class. Polymorphism is shown by the fact that the toString method defined in both Ork and Mork classes. This is also referred to as overriding. Overloading is represented by two constructors in either class with different parameter signatures.
27. Since the Comparable is implemented specifically for the Ork class, the parameter must be of type Ork, and not just Object.
28. Since the output of an object is controlled by the toString method, examine the toString method for each class and you will see the result. Also, since the nanu field is present in both classes, the toString method will use the nanu version that belongs to the object, but when outputting the nanu field directly, the one that belongs to the object reference is used. However, in the trey version, a curious thing happens – the compiler uses the value zero instead of the super class nanu value…interesting.
29. In the compareTo method arithmetic, the `one` object evaluates to 0 (1+2-3), and both the `two` and `trey` objects evaluate to 3 (1+2-0). Therefore the `one` object is less than both `two` and `trey` objects, and the `two` and `trey` objects are equal to each other.

30. The maximum value of the 16-bit short data type is 32767, which is 0111111111111111 in binary (0 + 15 1s), and is output without the leading zero.
31. The integer value 1000 takes 9 divisions by 2 to reach a value of 1….500, 250, 125, 62, 31, 15, 7, 3, 1.
32. See the Double class in the Java API for further clarification of the Double.toHexString method.
33. The contents of the list after the three fill statements is [0, 1, 2, 3, 3, 3, 3, 3, 2, 1], with a sum of 21.
34. In this replacement process, any "s" followed by another word character is replaced with "ss".
35. The Bubble and Insertion sorts each have an O(N) time complexity in the best case scenario, which is when the list is already sorted, or very nearly sorted.
36. Eight scenarios are rated at O(N^2) – Quicksort(worst), Bubble and Insertion(average and worst), Selection(all three scenarios)
37. O is 1 step away, N and T 2 steps, E and S 3 steps, and T 4 steps, for a total of 1+2+2+3+3+4 = 15
38. In the STATE tree, only A and T on level 1 have just one child, and in the CONTEST tree, C, N, T, and S have just one child, for a total of 6 nodes with just one child.
39. The remaining values in the first stack after the sequence (top to bottom) are 1 and 4, with 2 and 5 left in the second stack, 2 being the next one to be popped from the second stack.
40. The seven cycles in this directed graph are: GFG, GCG, CGDC, CFGC, CFGDC, ABCGDA, and ABCFGDA