University Interscholastic League

Computer Science Competition

Number 84 (Invitational B - 2004)

General Directions (Please read carefully!):

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

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

3) 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. You may 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. **All provided code segments are intended to be syntactically correct, unless otherwise stated. Ignore any typographical errors and assume any undefined variables are defined as used.**

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.

QUESTION 1

$47_8 * 22_8 =$

A. 1270_8 B. 1272_8 C. 1274_8 D. 1276_8 E. None of these

QUESTION 2

What replaces **<*1>** in the code to the right to indicate a block of code that should be executed once for the class?

A. `execute`

B. `once`

C. `static`

D. `final`

E. None of these

For the remaining questions, assume that **<*1>** has been filled in correctly.

QUESTION 3

What is output by the statement below?

```
System.out.print(
            Factorial.factorial(4));
```

A. nothing B. 0

C. 4 D. 24

E. None of these

QUESTION 4

Why are values being stored in the private variable `cache`?

A. To avoid recomputing factorials that have been previously computed, saving space.

B. To avoid recomputing factorials that have been previously computed, saving time.

C. It is impossible to compute factorials otherwise.

D. Every class has to have at least one data member.

E. None of these

```
// Class embeds function to compute
// factorials.  Uses the class BigInteger
// which represents integers of arbitrary
// length.  One BigInteger() constructor
// converts a String to a BigInteger.
// BigInteger.ZERO and BigInteger.ONE are
// class constants representing 0 and 1 as
// BigInteger objects.  The multiply()
// method multiplys a BigInteger by
// another.  The BigInteger class overrides
// the toString() method to convert a
// BigInteger to the corresponding string
// of digits.

import java.math.BigInteger;
import java.util.ArrayList;

public class Factorial {

  private static ArrayList cache;

  <*1> {
    cache = new ArrayList();
    cache.add(BigInteger.ONE);
  }

  public static BigInteger factorial(int n)
  {
    if (n<0) return BigInteger.ZERO;
    else {
      for (int i=cache.size(); i<=n; ++i)
      {
        BigInteger iBig = new BigInteger(
                          String.valueOf(i));
        cache.add(iBig.multiply(
              (BigInteger)cache.get(i-1)));
      }
      return (BigInteger)cache.get(n);
    }
  }
}
```

QUESTION 5

If a list contains 32 elements, what is the minimum number of comparisons that will be done when searching for an item in the list using the sequential search algorithm?

A. 0 B. 1 C. 5 D. 32 E. None of these

QUESTION 6

What is output by the method call `output("CATHY")`?

A. nothing
B. `cathy`
C. `CATHY`
D. `C`
E. None of these

QUESTION 7

What is output by the method call `output("Jane 3")`?

A. nothing
B. `Jane 3`
C. `JANE 3`
D. `J`
E. None of these

```
public static void output(String s) {
  for (int i=0; i<s.length(); ++i)
    System.out.print(
          Character.toUpperCase(
                s.charAt(i)));
}
```

QUESTION 8

What is output by the code to the right if `int y` has the value `14` before the loop begins?

A. `**********`
B. `*******`
C. `*****`
D. `**`
E. None of these

```
do {
  y=y-5;
  System.out.print('*');
} while (y>0);
```

QUESTION 9

Given `int a` and `float b`, which of these is a valid call to static method `f()`?

A. `f(a,b)`
B. `f(b,a)`
C. `f(b,b)`
D. `f(b,(double)a)`
E. More than one of these

```
public static int f(int x, double d) {
  // code not shown
}
```

QUESTION 10

What replaces **<*1>** in the code to the right to indicate that class `C` is a subclass of class `B`?

A. `subclass of class B`
B. `subclass of B`
C. `: public B`
D. `extends B`
E. None of these

```
public class C <*1> {
  // methods and data not shown
}
```

QUESTION 11

What replaces **<*1>** in the code to the right to give the number of elements of a?

A. `a.length-1`

B. `a.length`

C. `a.length+1`

D. `a`

E. None of these

```
public static void sort(int[] a) {
  for (int i=0; i < <*1>; ++i) {
    int min=a[i], minIndex=i;
    for (int j=i+1; j < <*1>; ++j)
      if (a[j]<a[minIndex]) {
        min=a[j]; minIndex=j;
      }
    a[minIndex]=a[i];
    a[i]=min;
  }
}
```

QUESTION 12

Assume that **<*1>** has been filled in correctly. Which sorting algorithm is being implemented?

A. `Selection` B. `Insertion`

C. `Mergesort` D. `Quicksort`

E. None of these

QUESTION 13

Which of the following data types can be used as the test condition for a `switch` statement?

A. `String` B. `char` C. `double` D. `Integer` E. None of these

QUESTION 14

What is returned by `mystery(15,3)`?

A. `0` B. `1`

C. `2` D. `3`

E. None of these

```
public static int mystery(int x, int y) {
  if (x%y == 0) return y;
  else return mystery(y, x%y);
}
```

QUESTION 15

What is returned by `mystery(48,80)`?

A. `8` B. `48`

C. `16` D. `80`

E. None of these

QUESTION 16

What is returned by `Double.parseDouble("24.3e-3")`?

A. `24.3` B. `21.3` C. `0.0243` D. Exception thrown E. None of these

QUESTION 17

What is returned by `process("abcde","3b")`?

A. `abbcde`
B. `babcde`
C. `abcbde`
D. `bbbbbb`
E. None of these

QUESTION 18

What is returned by `process("abcd","3b3c3d2e")`?

A. `abecdcbd`
B. `abcdbbbcccdddee`
C. `bbbcccdddeeabcd`
D. `abecbcdd`
E. More than one of these

```
public String process(String s, String t) {
  StringBuffer sb = new StringBuffer(s);
  int i=0;
  while (i<t.length())
    sb.insert(t.charAt(i++)-'0',
              t.charAt(i++));
  return sb.toString();
}
```

QUESTION 19

What is returned by `comparison(27,27)`?

A. `729` B. `54`
C. `0` D. `27`
E. None of these

QUESTION 20

What is returned by `comparison(15,27)`?

A. `729` B. `54`
C. `0` D. `27`
E. None of these

```
public static int comparison(int a, int b){
  int and = a&b;
  int or = a|b;
  int xor = a^b;

  return (and>or)?((and>xor)?and:xor):
                  ((or>xor)?or:xor);
}
```

QUESTION 21

What replaces **<*1>** in the code to the right to make the `while` loop immediately terminate when `test` is `false`, with execution continuing with the first line of code immediately after the loop?

A. `stop;`
B. `continue;`
C. `throw new Exception();`
D. `break;`
E. None of these

```
boolean test=true;

while (true) {
  // code not shown
  if (!test) <*1>
  // more code not shown
}

// more code not shown
```

QUESTION 22

What does `int[][] intArray` look like after executing the code to the right?

```
int[][] intArray = new int[3][3];

for (int i=0; i<3; ++i)
  for (int j=0; j<3; ++j)
    intArray[j][i] = j*i+j;
```

A.

0	0	0
0	0	0
0	0	0

B.

0	0	0
1	2	3
2	4	6

C.

0	1	2
0	1	2
0	1	2

D.

0	1	2
1	2	3
2	3	4

E. None of these

QUESTION 23

What does `int[] a` look like after the static method call `turn(a, 2, 5)` if a starts as the array below?

1	2	3	4	5	6

```
public static void turn(int[] a, int begin,
                                 int end) {
  int temp, diff=end-begin;

  for (int i=0; i<=diff/2; ++i) {
    temp=a[begin+i];
    a[begin+i]=a[end-i];
    a[end-i]=temp;
  }

}
```

A.

1	2	3	4	5	6

B.

1	2	6	5	4	3

C.

1	2	6	4	5	3

D.

6	5	4	3	2	1

E. None of these

QUESTION 24

Which of the following types is stored in the largest number of bits?

A. `boolean` B. `short` C. `int` D. `long` E. `char`

QUESTION 25

What interface name replaces **<*1>** in the code to the right so that the `BST` methods can use the `compareTo()` method on data?

A. `Object`

B. `Serializable`

C. `Cloneable`

D. `Comparable`

E. None of these

For the remaining questions, assume that **<*1>** has been filled in correctly.

QUESTION 26

When a `BST` object is constructed, what value will `left` and `right` be initialized to?

A. `0` B. `null`

C. `this` D. `undefined`

E. None of these

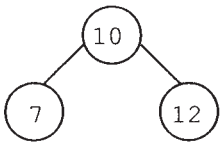
QUESTION 27

What does `BST b` look like after executing the code below?

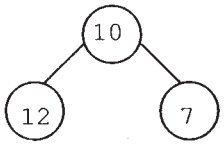
```
BST b = new BST (new Integer(10));
b.add(new Integer(7))
 .add(new Integer(12));
```

A. Syntax error in code

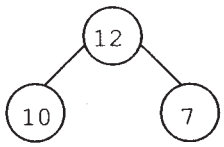
B.



C.



D.



E. More than one of these

```
public class BST {

  public BST(<*1> value) {
    data=value;
  }

  public BST add(<*1> value) {
    int c = data.compareTo(value);

    if (c<0) {
      if (right!=null)
        right.add(value);
      else right = new BST(value);
    }

    else if (c>0) {
      if (left!=null)
        left.add(value);
      else left = new BST(value);
    }

    return this;
  }

  // other methods not shown

  private BST left, right;
  private <*1> data;
}
```

QUESTION 28

Which of the following makes an `Employee` named Jane Doe with social security number 111-11-1111 and starting salary 2365.42?

A.
```
new Employee(Jane Doe,
                    111-11-1111, 2365.42)
```

B.
```
new Employee("Jane Doe",
                    111-11-1111, 2365.42)
```

C.
```
new Employee("Jane Doe",
                    "111111111", 2365.42)
```

D.
```
new Employee("Jane Doe",
                    111111111, 2365.42)
```

E. More than one of these

QUESTION 29

If `Employee e` is set to the employee created above, what is returned by `e.level()`?

A. 0 B. 1

C. 2 D. 3

E. None of these

QUESTION 30

Given `Employee[] workers`, which of these gives a $100 per month raise to each employee in the array at level 3 or less?

A.
```
for(int i=0; i<workers.length; ++i)
  if(workers[i].level<3)
    workers[i].raise(100);
```

B.
```
for(int i=0; i<workers.length; ++i)
  if(workers[i].level<=3)
    workers[i].raise(100);
```

C.
```
for(int i=0; i<workers.length; ++i)
  if(workers[i].level()<3)
    workers[i].raise(100);
```

D.
```
for(int i=0; i<workers.length; ++i)
  if(workers[i].level()<=3)
    workers[i].raise(100);
```

E. More than one of these

```
public class Employee {

  public Employee(String name, long SSN,
                  double monthlySalary) {
    this.name=name;
    this.SSN=SSN;
    this.monthlySalary=monthlySalary;
  }

  public int level() {
    return (int)monthlySalary/1000 + 1;
  }

  public void raise(double extra) {
    monthlySalary+=extra;
  }

  private String name;
  private long SSN;
  private double monthlySalary;
}
```

QUESTION 31

What replaces **<*1>** in the code to the right so that an object implementing the `Set` interface is created?

A. `new Set()`

B. `new TreeSet()`

C. `new List()`

D. `new TreeList()`

E. More than one of these

```
public static int count(List list) {
  Set s = <*1>;
  Iterator i = list.iterator();

  while (i.hasNext())
    s.add(i.next());

  return s.size();
}
```

QUESTION 32

Assume that **<*1>** has been filled in correctly. If `List myList` contains, in order, the elements below (represented as `Integer` objects) what is returned by the static method call `count(myList)`?

```
13 24 10 28 13 24 27 78
```

A. 5 B. 6

C. 7 D. 8

E. None of these

QUESTION 33

What replaces **<*1>** in the code to the right so that `d1` and `d2` are each randomly assigned a number from the set `{1, 2, 3, 4, 5, 6}`?

A. `r.nextInt(5) + 1`

B. `r.nextInt(6) + 1`

C. `r.nextInt(7) + 1`

D. `r.nextInt(1,6)`

E. None of these

```
public static int rollDice() {
  Random r = new Random();

  int d1 = <*1>, d2 = <*1>;

  return d1+d2;
}
```

QUESTION 34

If class `Child` is a subclass of class `Parent`, what is the syntax for calling a private method of `Parent` named `f()` from within a private method of `Child`?

A. `this.f()` B. `f()` C. `super.f()` D. `super(f())` E. None of these

QUESTION 35

What is output by the code to the right if `ch` is `'z'`?

A. `default`
B. syntax error since `A`, `B`, and `C` are not literals
C. nothing
D. exception thrown
E. None of these

```
final char A = 'A', B = 'B', C = 'C';

switch(ch) {
case A: System.out.print("Case A");
        break;
case B: System.out.print("Case B");
        break;
case C: System.out.print("Case C");
        break;
}
```

QUESTION 36

What is returned by the static method call `nand(3<7, 5>=17)`?

A. `false`
B. `true`
C. `1`
D. `0`
E. None of these

```
public static boolean nand(boolean a,
                           boolean b) {
  return !(a && b);
}
```

QUESTION 37

Which of these expressions could have been used as the return value for `nand()` without changing its behavior?

A. `!a && !b`
B. `!(a || b)`
C. `a || b`
D. `!a || !b`
E. None of these

QUESTION 38

What is output by the code to the right?

A. `"+"+`
B. `\+\"`
C. `\"+"+"\"+`
D. Syntax error
E. None of these

```
System.out.print("\"+"+"\"+");
```

QUESTION 39

Assume `getInt()` returns the next `int` from user input. What is the value of sum after executing the code to the right on the input below?

```
4 3 2 1 0 -1
```

A. `1000`
B. `98`
C. `10`
D. `0`
E. None of these

```
int i=0, sum=0;

while(i!=-1) {
  i = getInt();
  sum += sum + i;
}
```

QUESTION 40

What could the type of `z` be for the call to `add()` to work?

A. `int`
B. `boolean`
C. `Float`
D. All of these
E. None of these

```
ArrayList a = new ArrayList();

// some code left out

a.add(z);
```

COMPUTER SCIENCE ANSWER KEY
UIL INVITATIONAL B 2004

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

IMPORTANT NOTE TO GRADERS: Correct answers receive **6 points**, and incorrect answers receive a deduction of **2 points**. No points are given or deducted for unanswered questions.