1. Short Answer

1. Write the value of (double) \( \frac{22}{7} \).

2. Write this number using Java’s scientific notation, (without using Math.pow):
   \( 6.022 \cdot 10^{23} \)

3. True or False. Any for loop can be converted into an equivalent while loop.

4. True or False. Any recursive method can be re-written as a non-recursive method using loops.

5. True or False. The following condition will compile in Java.
   \((a < b < c)\)

6. You have a program called Recipe.java which reads from standard input and writes to standard output. You have compiled it. The command-line to run it so it reads keyboard input and writes to the terminal screen is: java Recipe.
   Write the command-line to run it so it reads input redirected from a file named cookbook.txt.
   Write the command-line to run it so it reads input from cookbook.txt and writes to an output file named meal.txt.
   Write the command-line to run it so it reads keyboard input and pipes the output to another compiled program named HungryThing.java.
2. Arrays, Functions, Analysis of Algorithms (from Fall05 Exam 1, question 6)

The following two methods do the same job. They each take an ORDERED array of ints and a target number as arguments.

```java
public static boolean mystery1(int[] array, int target) {
    for (int i = 0; i < array.length; i++) {
        if (array[i] == target) return true;
        else if (array[i] > target) return false;
    }
    return false;
}

public static boolean mystery2(int[] array, int target) {
    int low = 0;
    int high = array.length - 1;
    while (low <= high) {
        int mid = (low + high) / 2;
        if (array[mid] == target) return true;
        else if (array[mid] < target) low = mid + 1;
        else high = mid - 1;
    }
    return false;
}
```

Use this array to answer the following questions:

```java
int[] a = { 2, 5, 11, 14, 15, 27, 31};
```

a) What does mystery1(a, 5) return?

b) Fill in the trace table to show that mystery2(a, 5) returns the same thing.

<table>
<thead>
<tr>
<th>target</th>
<th>low</th>
<th>hi</th>
<th>mid</th>
<th>return value</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>31</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
c) What do `mystery1(a, 20)` and `mystery2(a, 20)` return?

d) In general, what do these methods do?

e) How many comparisons with the target will `mystery1(a, 32)` make?

f) How many comparisons with the target will `mystery2(a, 32)` make?

g) Which one would you use for an array of a million elements?

h) Why?
3. Recursion, Debugging (from Spring04, Exam 1, Question 4)

There's a bug in the following recursive program. You need to find it and fix it.

```java
public class Series{
    public static int func(int j){
        if (j==1) return 1;
        return 2*func(j-1)+5*func(j-2);
    }

    public static void main(String[] args) {
        int N=Integer.parseInt(args[0]);
        if (N<0) {
            System.out.println(''invalid argument'');
            return;
        }
        System.out.println(func(N));
    }
}
```

a. Draw the recursion tree for func(3). You only need to draw the tree up to 3 levels, which means the height of the recursion tree should be no greater than 3.

b. From the recursion tree in (a), do you see a problem with the program? Explain what is the problem.

4. Performance. The following table gives approximate running times for a program with N inputs for various values of N.

<table>
<thead>
<tr>
<th>N</th>
<th>time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>5 seconds</td>
</tr>
<tr>
<td>2000</td>
<td>20 seconds</td>
</tr>
<tr>
<td>5000</td>
<td>2 minutes</td>
</tr>
<tr>
<td>10000</td>
<td>8 minutes</td>
</tr>
</tbody>
</table>

Which of the following best describes the likely running time of this program for $N = 100,000$?

V. A few minutes
W. A few hours
X. Half a day
Z. A few days