CS 151 **Exam-2** **April 8, 2004**

This Quiz is being given under the guidelines of the **Honor Code**. You are expected to respect those guidelines and to report those who do not. Answer the questions in the spaces provided. If you run out of room for an answer, continue on the back of the page. There are 6 questions for a total of 85 points.

Name: _____

1. (5 points) What is the running time for the following algorithm?

```
1  public static boolean isPrime( int n ) {
2     if( n == 2 || n == 3 )
3        return true;
4     if( n % 2 == 0 )
5       return false;
6     for( int i = 3; i <= squareRoot( n ); i+=2 )
7       if( n % i == 0 )
8          return false;
9     return true;
10 }
```

2. (15 points) Replace the loop in lines 6-9 with a recursive helper function. *Hint:* The helper should take two parameters.

3. Given the following recursive function:

```
1  private void mystery( char [ ] str, int low, int high )      {
2      if( low < high )
3          println( str );
4      for( int i = low; i <= high; i++ )          {
5          char [ ] tmp = str.clone( ); // tmp will be str
6          tmp[ i ]    = str[ low ];       // with i and low
7          tmp[ low ] = str[ i ];          // swapped
8          mystery( tmp, low + 1, high );
9      }
10 }
```

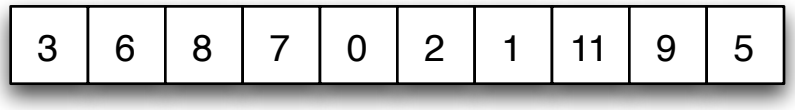
(a) (20 points) Trace the calls for mystery when it is called as follows: `mystery("abc", 0, 2);`

(b) (5 points) what is the running time for mystery in big O terms?

(c) (5 points **extra credit**) Rewrite mystery so it is not recursive.

4. (20 points) Write an $O(n \log n)$ algorithm that will remove all duplicates from an array of n floats. You may use Arrays.sort(), but no other Classes from the Java Collections library.

5. (10 points) Using the *Median of 3* approach, show how quicksort would parition the following array of numbers. When you are done make sure that you clearly identify L, R, and the pivot.



6. (10 points) Given the following array $int[]a = \{90, 1, 3, 8, 56, 21, 5, 13, 1, 146, 2, 34\}$. Show the contents of a during shell sort, after a gap size of 5, 2, and 1. **Note Bene** You do not need to show every swap, just the contents after the processing for each gap size.