Name: _____

This exam is open book, open notes, but there can be no sharing of any material. You can use the Internet, but only as a library. If you are not sure if something is allowed, please ask. If you use any material other than the assigned readings and your own prior work, you must prominently indicate the source in your answers. There are 5 questions on 6 pages. You have *40 minutes* to earn *40 points*. You may wish to use this correspondence to plan your time.

1. (1 pt.) Write your name in the space provided above.

2. (8 pts.) The parenthetical remark in the last paragraph on page 761 of Thompson's paper[1] suggests that output of the program in Figure 1 in that paper differs from the program's listing in Figure 1. Explain this difference as precisely as you can. (Assume that the text "`(213 lines deleted)`" has been replaced by the appropriate text suggested by the comment in the listing, and ignore the explanatory material following the last closing brace.) Hint: There are differences other than white-space differences.

[1] Ken Thompson, "Reflections on Trusting Trust," *Communications of the ACM* 27/8 (1984).

3. (15 pts.) Suppose file `c1.c` contains the complete source code of a typical C compiler, as suggested by Figure 2.2 in Thompson's paper.[2] Suppose `c2.c` is a modified version of `c1.c` in which the special character code `\v` is implemented as suggested by Figure 2.1. Similarly, suppose file `c3.c` contains a modified version of `c1.c` in which the character code `\v` is implemented as suggested by Figure 2.3.

   Let $E(c, p)$ be the result (executable code) of compiling file p (source code) with compiler c (executable code). If compiler c generates a compile-time error on input p then $E(c, p)$ is undefined.

   Assume that an executable $\texttt{c1} = E(\texttt{c1}, \texttt{c1.c})$ is initially available.

   For each of the following compilations, indicate whether there is any error, *explaining each answer briefly.*

   (a) $E(\texttt{c1}, \texttt{c2.c})$

   (b) $E(\texttt{c1}, \texttt{c3.c})$

   (c) $E(E(\texttt{c1}, \texttt{c2.c}), \texttt{c1.c})$

   (d) $E(E(\texttt{c1}, \texttt{c2.c}), \texttt{c2.c})$

   (e) $E(E(\texttt{c1}, \texttt{c2.c}), \texttt{c3.c})$

   (f) $E(E(\texttt{c1}, \texttt{c3.c}), \texttt{c1.c})$

   (g) $E(E(\texttt{c1}, \texttt{c3.c}), \texttt{c2.c})$

   (h) $E(E(\texttt{c1}, \texttt{c3.c}), \texttt{c3.c})$

   (i) $E(E(E(\texttt{c1}, \texttt{c3.c}), \texttt{c2.c}), \texttt{c2.c})$

   (j) $E(E(E(E(\texttt{c1}, \texttt{c3.c}), \texttt{c2.c}), \texttt{c2.c}), \texttt{c3.c})$

[2] *Idem*, Figures 2.1 and 2.2 appear out of order in the paper.

[additional space for answering the question on the previous page]

4. (8 pts.) Based on Bracha's tutorial[3] of generics in Java, find all (any) statements that generate compile-time errors in the following code fragment. *Explain your answers.*

   Treat each error independently. That is, when you find an error, analyze the rest of the program by assuming the statement containing the error has been removed.

```
public class Apple extends Fruit { }  // in file Apple.java
public class Orange extends Fruit { }  // in file Orange.java

// file Fruit.java:
import java.util.List;
import java.util.ArrayList;
public class Fruit {
    public static void m1(List<Fruit> a) { }
    public static <T> void m2(List<T> a) { }
    public static void m3(List<? extends Fruit> a) { }
    public static void m4(List<?> a) { }
    public static <T extends Fruit> void m5(List<T> a) { }

    public static void main(String[] args) {
       List<Fruit> s1 = new ArrayList<Fruit>();
       List<Apple> s2 = new ArrayList<Apple>();
       List<Orange> s3 = new ArrayList<Orange>();
       List<Orange> s4 = s2;
       List<Fruit> s5 = s2;
       m1(s1); m1(s2); m1(s3);
       m2(s1); m2(s2); m2(s3);
       m3(s1); m3(s2); m3(s3);
       m4(s1); m4(s2); m4(s3);
       m5(s1); m5(s2); m5(s3);
   }
}
```

[3]Gilad Bracha, Generics in the Java Programming Language, Tutorial. `http://java.sun.com/`, 2004.

[additional space for answering the question on the previous page]

5. (8 pts.) Define an *RGB-graph* to be a simple undirected graph (no loops or multi-edges) in which each edge is associated with one of three colors: red, green, and blue.

   What is the number of distinct RGB-graphs on the set $[n] = \{1, 2, \ldots, n\}$ of vertices? Express your answer as an expression that contains n, making it as compact as possible. *Justify your answer.*