Nome: _____ Matrícula: _____

1. O objetivo desta questão é desenvolver uma estratégia de teste de software para validar a interface que colore grafos dada abaixo:

```
import java.util.List;
public interface ColorableGraph {
  public void link(String s1, String s2);
  public void setColor(String s1, int color);
  public int getColor(String s2);
  public List<String> getNeighbors(String node);
}
```

Uma coloração de um grafo G é uma função C que, para cada nodo $n \in G$ retorna um inteiro $i, i > 0$. C é uma coloração válida de G se, e somente se, para cada par de nodos $n_1 \in G$ e $n_2 \in G$, se n_1 for adjacente a n_2 então $C(n_1) \neq C(n_2)$. C é uma coloração ótima se para cada nodo n, $C(n)$ é igual ou menor ao número cromático de G. O número cromático de um grafo é o menor número de cores necessário para colorir o grafo. Eu comecei a construir um programa de testes, o qual pode ser visto logo abaixo:

```
import junit.framework.TestCase;
import junit.swingui.TestRunner;
public class TestGraph extends TestCase {
  private ColorableGraph cg = new ColorableGraphStub();
  public static void main(String[] args) {
    TestRunner.main(new String[] { "Test Graph Coloring" });
  }
  // Esta funcao sera sempre chamada antes dos testes comecarem.
  public void setUp() throws Exception {
    cg.link("a", "b");
    cg.link("a", "c");
    cg.link("d", "c");
  }
  public void testCliqueColoring() throws Exception {
    cg.setColor("a", 1);
    assertEquals(cg.getColor("a"), 1);
  }
}
```

`junit` sempre irá invocar a função `setUp`, e a seguir cada uma das funções que tem a assinatura `public void test* throws Exception`. Escreva mais três funções de teste, explicando o que estes testes buscam validar. Você pode modificar `setUp` conforme necessário.