1 - Qual é a sequência de letras ao executar a classe de teste Primeiro?
R: d, b, f, a, b, g, a, c

2 - Por que os métodos anotados com @BeforeClass e @AfterClass precisam ser estáticos?
R: porque são executados antes do construtor da classe.

3 - Por que o método b(), da classe de teste Segundo, apresenta um resultado azul e o método c() um resultado verde?
R: porque ambos esperam uma Exception, mas o b retorna false, ou seja, não retorna exception.

4 - O método e(), da classe de teste Segundo, apresenta um resultado vermelho. Qual é a alteração necessária no método de teste e() para que o resultado seja verde?
R: uma das soluções é setar o primeiro parametro pra 1, e setar o primeiro valor da função "a.n()" para 1;

5 - Programar um método de teste para testar o método o(), da classe A. O resultado do teste deverá ser verde.
R:
```
@Test
	public void o(){
		a.o();
	}
```

6 - Programar um método de teste para testar o método p(), da classe A. O resultado do teste deverá ser verde. Observação: use o método assertSame da classe org.junit.Assert.

```
@Test
public void p(){
	String aux = "oi";
	assertSame(aux, a.p());
}
```

7 - Qual é o motivo dos métodos assertEquals e assertSame produzirem resultados diferentes nos teste dos métodos f() e g() da classe Segundo?
R: assertEquals compara 2 objetos, enquanto que assertSame compara a referecia passada do objeto.