Two-coloring a graph

You are given an *undirected and connected* graph $G = (V, E)$. Provide an algorithm that colors the vertices in V with two colors R(red) and B(blue), such that no neighbors $(u, v) \in E$ has the same color; if this *is not possible*, then the algorithm should print that it is impossible.

You may assume that each vertex in V has a field $v.\texttt{color}$ that is initially set to `null`, and that you are provided with a function `FlipColor`, where `FlipColor(R)` $= B$ and `FlipColor(B)` $= R$.

Input: A connected graph $G = (V, E)$
Output: A graph where the vertices are colored, if it is possible

```
1 Procedure TwoColor(G)
    // ···
```
