```
1  using System.Collections.Generic;
2  using UnityEngine;
3
4  /// <summary>
5  /// Logical representation of an INPUT gate.
6  /// </summary>
7  public class InputGate : Circuit
8  {
9      /// <summary>
10     /// Powered status unique to an INPUT gate.
11     /// </summary>
12     private bool powered;
13
14     public InputGate() : this(Vector2.zero) { }
15
16     public InputGate(Vector2 startingPos) : base("INPUT", 0, 1,
         startingPos) { }
17
18     /// <summary>
19     /// Returns an output to update if the output has changed due to
         alterations in input power statuses.
20     /// </summary>
21     /// <returns>The list of outputs that should have their connections
         called.</returns>
22     protected override List<Output> UpdateOutputs()
23     {
24         bool outputStatus = Outputs[0].Powered;
25         List<Output> outputs = new List<Output>();
26
27         // INPUT gate representation
28         Outputs[0].Powered = powered;
29
30         if (outputStatus != Outputs[0].Powered) outputs.Add(Outputs[0]);
31
32         return outputs;
33     }
34
35     /// <summary>
36     /// Getter and setter method; setting the powered value of the input
         node will also create an update call.
37     /// </summary>
38     public bool Powered { get { return powered; } set { powered = value;
         Update(); UpdateChildren(); } }
39 }
```