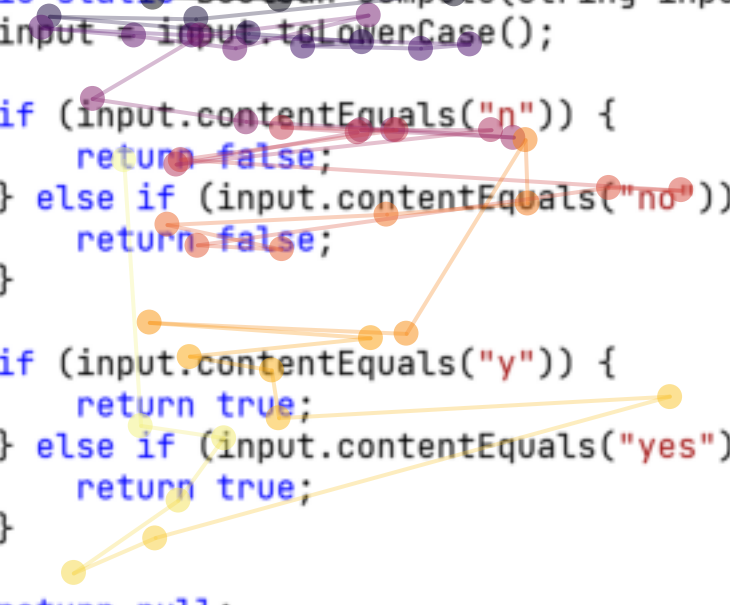


```
public static Boolean compute(String input) {  
    input = input.toLowerCase();  
  
    if (input.contentEquals("n")) {  
        return false;  
    } else if (input.contentEquals("no")) {  
        return false;  
    }  
  
    if (input.contentEquals("y")) {  
        return true;  
    } else if (input.contentEquals("yes")) {  
        return true;  
    }  
  
    return null;  
}
```



The image displays a control flow graph (CFG) overlaid on a Java code snippet. The graph consists of nodes (colored circles) and edges (colored lines) representing the execution flow. The nodes are color-coded: purple for the initial assignment and the first if-else block, red for the 'no' branch, orange for the 'y' and 'yes' branches, and yellow for the final return statements. The flow starts at the top, goes through the assignment, then branches based on the input string. It follows the 'no' path (red), the 'y' path (orange), or the 'yes' path (yellow) before reaching the final return null statement.