

```
public static boolean compute(String word) {  
    boolean result = true;  
  
    for (int i = 0, j = word.length() - 1; i < word.length() / 2; i++, j--) {  
        if (word.charAt(i) != word.charAt(j)) {  
            result = false;  
            break;  
        }  
    }  
  
    return result;  
}
```

The diagram illustrates data flow between lines of code. Nodes are represented by colored circles: purple for 'boolean' and 'result', orange for 'i' and 'j', yellow for 'word.length()', and grey for 'true'. Lines connect these nodes across different lines of code, showing how variables and values are passed and updated. For example, 'result' is initialized to 'true' and then potentially updated to 'false' within the loop. 'i' and 'j' are initialized and then incremented/decremented in the loop. 'word.length()' is used to calculate the loop bounds and the string length for character access.