**Recipe name:** simple_poker

**Inputs:**

*input_hand1*, a 5-tuple of integers representing the first hand

*input_hand2*, a 5-tuple of integers representing the second hand

**Outputs:**

either 0, 1, or 2 depending on which hand wins or if there is a tie

1. Let *hand1* be an empty mapping
2. Let *hand2* be an empty mapping
3. Let *score1*, *score2*, be integers equal to 0
4. Let *current* be an integer equal to 1
5. Let *comparison1*, *comparison2* be empty sequences
6. For each *card* in *input_hand1*, do the following:
    a. If *card* is a key in *hand1*, then
        i. Add 1 to the value corresponding to the key *card* in *hand1*
    b. Otherwise,
        i. Set the value corresponding to the key *card* in *hand1* to 1
7. For each *card* in *input_hand2*, do the following:
    a. If *card* is a key in *hand2*, then
        i. Add 1 to the value corresponding to the key *card* in *hand2*
    b. Otherwise,
        i. Set the value corresponding to the key *card* in *hand2* to 1
8. If the length of *hand1* is less than the length of *hand2*,
    a. Return 1
9. If the length of *hand2* is less than the length of *hand1*,
    a. Return 2
10. For each *key1*, *value1* in *hand1*, do the following
    a. If *value1* is 4, then
        i. Add 5 points to *score1*
        ii. *current* ← 4
    b. If *value1* is 3, then
        i. Add 3 points to *score1*
        ii. If *current* < 4, then
            1. *current* ← 3
    c. If *value1* is 2, then
        i. Add 1 point to *score1*
        ii. If *current* < 3, then
            1. *current* ←2
11. For each *key2*, *value2* in *hand2*, do the following
    a. If *value2* is 4, add 5 points to *score2*
    b. If *value2* is 3, add 3 points to *score2*

    c. If *value2* is 2, add 1 point to *score2*
12. If *score1* > *score2*, then
    a. Return 1
13. If *score2* > *score1*, then
    a. Return 2
14. If *score1* is equal to *score2*, then
    a. While *current* > 0, do the following
        i. For each *key1*, *value1* in *hand1*, do the following
            1. If *value1* = *current*, then
                a. Append the value *key1* to the end of *comparison1*
        ii. For each *key2*, *value2* in *hand2*, do the following
            1. If *value2* = *current*, then
                a. Append the value *key2* to the end of *comparison2*
        iii. Sort *comparison1* and *comparison2* so that they go from the highest value to the lowest
        iv. For each number *idx* from 0 to the length of *comparison1* – 1, do the following
            1. If the value at the index *idx* for *comparison1* is greater than the value at the index *idx* for *comparison2*, then
                a. Return 1
            2. If the value at the index *idx* for *comparison2* is greater than the value at the index *idx* for *comparison1*, then
                a. Return 2
        v. Set *comparison1* and *comparison2* to empty sequences
        vi. *current* ←*current* – 1
15. Return 0