**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:
   1. If *card* is a key in *hand1*, then
      1. Add 1 to the value corresponding to the key *card* in *hand1*
   2. Otherwise,
      1. Set the value corresponding to the key *card* in *hand1* to 1
7. For each *card* in *input\_hand2*, do the following:
   1. If *card* is a key in *hand2*, then
      1. Add 1 to the value corresponding to the key *card* in *hand2*
   2. Otherwise,
      1. 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*,
   1. Return 1
9. If the length of *hand2* is less than the length of *hand1*,
   1. Return 2
10. For each *key1*, *value1* in *hand1*, do the following
    1. If *value1* is 4, then
       1. Add 5 points to *score1*
       2. *current* 🡨 4
    2. If *value1* is 3, then
       1. Add 3 points to *score1*
       2. If *current* < 4, then
          1. *current* 🡨 3
    3. If *value1* is 2, then
       1. Add 1 point to *score1*
       2. If *current* < 3, then
          1. *current* 🡨2
11. For each *key2*, *value2* in *hand2*, do the following
    1. If *value2* is 4, add 5 points to *score2*
    2. If *value2* is 3, add 3 points to *score2*
    3. If *value2* is 2, add 1 point to *score2*
12. If *score1* > *score2*, then
    1. Return 1
13. If *score2* > *score1*, then
    1. Return 2
14. If *score1* is equal to *score2*, then
    1. While *current* > 0, do the following
       1. For each *key1*, *value1* in *hand1*, do the following
          1. If *value1* = *current*, then
             1. Append the value *key1* to the end of *comparison1*
       2. For each *key2*, *value2* in *hand2*, do the following
          1. If *value2* = *current*, then
             1. Append the value *key2* to the end of *comparison2*
       3. Sort *comparison1* and *comparison2* so that they go from the highest value to the lowest
       4. 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
             1. Return 1
          2. If the value at the index *idx* for *comparison2* is greater than the value at the index *idx* for *comparison1*, then
             1. Return 2
       5. Set *comparison1* and *comparison2* to empty sequences
       6. *current* 🡨*current* – 1
15. Return 0