Given a template for a program that evaluates Poker Hands, students implement a 2-LEVEL TALLYING ALGORITHM that UNIQUELY distinguishes between 1 Pair, 2 Pair, 3-of-a-Kind, Full House and 4-of-a-Kind.

How does the Tally method work?

When you tally (count) items, the VALUE of the item is used as an INDEX into the tally list.

For example, when you count ballots, a candidate ID NUMBER is used as an INDEX into the Tally list, specifying the slot where the votes for that candidate get counted.

In Poker, each hand consists of 5 cards, and the RANK of each card is a value in the range 1 (Ace) to 13 (King). To tally a hand, you create a 14-slot list. The RANK of a card is used as the INDEX specifying the slot in the TALLY list (the green columns in Figure 1). When counted, the values in the Tally list show how many times each card rank appears in the hand.

The next step is to tally the 13 slots (1-13) of the Tally list. We do this by creating a 2nd MULTIPLES Tally list with 5 slots. The VALUE of each member of the 1st Tally list is now used as an INDEX into a corresponding slot in the MULTIPLES list (the pink columns). This shows how many single cards, pairs, triplets, etc. appear in the hand.

In this way, Poker Hands can be distinguished from one another using simple one-line BOOLEAN expressions. For example:

1 PAIR: \[\text{multiples}[2] == 1 \&\& \text{multiples}[3] == 0 \text{ or } \text{multiples}[2] == 1 \&\& \text{multiples}[1] == 3\]

2 PAIR: \[\text{multiples}[2] == 2\]

3-of-a-KIND: \[\text{multiples}[3] == 1 \&\& \text{multiples}[2] == 0 \text{ or } \text{multiples}[3] == 1 \&\& \text{multiples}[1] == 2\]

FULL HOUSE: \[\text{multiples}[3] == 1 \&\& \text{multiples}[2] == 1\]

4-of-a-KIND: \[\text{multiples}[4] == 1\]