Count: 0 — Preference
4398762A10JQK5

Count: 1 — Preference
A10JQK87653294

Count: 2 — Bonus
~~A~~ 2A
2A10JQK97654 83

Count: 3 — Bonus
~~A~~ 3A
A10JQK98654372

Count: 4 — Bonus
~~2 3 A~~ 432
10JQK98754326A

Count: 5 — Bonus
10JQK5432A
10JQK98764 32A5

Count: 6 — Bonus
~~A 2 3 4 5 6 9~~ 965432A
910JQK876532A4

Count: 7 — Bonus
~~8 A 2 3 4 5 6 7 8~~ 8765432A
810JQK976542A3

Count: 8 — Bonus
8765432A
710JQK986543A2

In Bonus Calc, 21 count is worth -2 points.

Count: 9 - Bonus
98765432
610JQK9875432A

Count: 10 - ~~Bonus~~ All-copy Bonus
510JQK9876432A

Count: 11 - ~~Bonus~~ All-copy Bonus
~~98765432A~~
49876532A10JQK

Count: 12 - ~~Bonus~~ All-copy Bonus
~~10JQK87654 32A~~
~~310JQK87654 2A9~~ 3876541 0JQK2A9

Count: 13 - ~~Bonus~~ All-copy Bonus
~~10JQK9765432A~~
~~210JQK976543A8~~ 2765431 0JQK9A8

Count: 14 - ~~Bonus~~ All-copy Bonus
~~10JQK98654 32A~~
~~A10JQK9865432~~ A65432 10JQK987

Count: 15 - ~~Bonus~~ All-copy Bonus
~~10JQK98754 32A~~
~~10JQK98754 32A6~~ 5432A10JQK9876

Count: 16 - ~~Bonus~~ All-copy Bonus
~~10JQK98764 32A~~
~~10JQK98764 32A3~~ 432A10JQK98765

Count: 17 - Bonus
All-copy
~~10JQK98765 32A~~
32A10JQK987654

In Bonus Calc, 21 count is worth -2 points.

Count: 18 - ~~Bonus~~ All-copy Bonus
~~10JQK9876543A~~
2A10JQK9876543

Count: 19 - ~~Bonus~~ All-copy Bonus
~~10JQK98765432A~~
A10JQK98765432

Count: 20 - ~~Bonus~~ All-copy Bonus
~~10JQK98765432~~
10JQK98765432A

Count: 21 - All-Copy Bonus
10JQK98765432A

Count: 22 - All-Copy Bonus
98765432A

Count: 23 - All-copy Bonus
8765432A

Count: 24 - All-copy Bonus
765432A

Count: 25 - All-Copy Bonus
65432A

Count: 26 - All-Copy Bonus
5432A

Count: 27 - All-Copy Bonus
432A

Count: 28 - All-Copy Bonus
32A

Count: 29 - All-Copy Bonus
2A

Count: 30 - ~~All-copy Bonus~~
A

↑ Wait, no, Preference type! Save processing time.

- myTurn = true (initialize)
  If(!dealer) myTurn = false
- Arraylist<PlayingCard> hand (initialize)
  ← fill this with cards chosen earlier
- ArrayList<PlayingCard> history (initialize)
  ← starts empty
- int counter = 0 (initialize)
- ArrayList<PlayingCard> opponent (initialize)
  ← Starts empty

```
while(hand.size()>0 || opponent.size()<4){
    if(myTurn){
        if(hand.empty()){
            myTurn = !myTurn
            continue
        }
selection = pegging.get(counter).exec(hand, history)
        if(selection == null){
            // opponent plays untill they can't.
            // log opponent plays.
            // Award points as necessary
            // Reset counter and Reset history
            // flip myTurn ONLY if opponent could play
        }else{
            // log selection
            // Award points as necessary
            // Remove selection from hand
            // Add selection to history, update counter
        }
        myTurn = !myTurn
    continue;
} else{
    // OPPONENT PLAY
```