Jim is writing a program for statistically analyzing card games. He needs to store many different card hands in memory efficiently. Each card has one of four *suits* and one of thirteen *values*. In his implementation, each hand is stored as a linked list of cards in a canonical order: the cards are first ordered by suit: all the clubs come first, followed by all the diamonds, then all the hearts, and finally the spades. Within each suit, the cards are ordered by value: A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K. Each hand contains at most one of any given card.
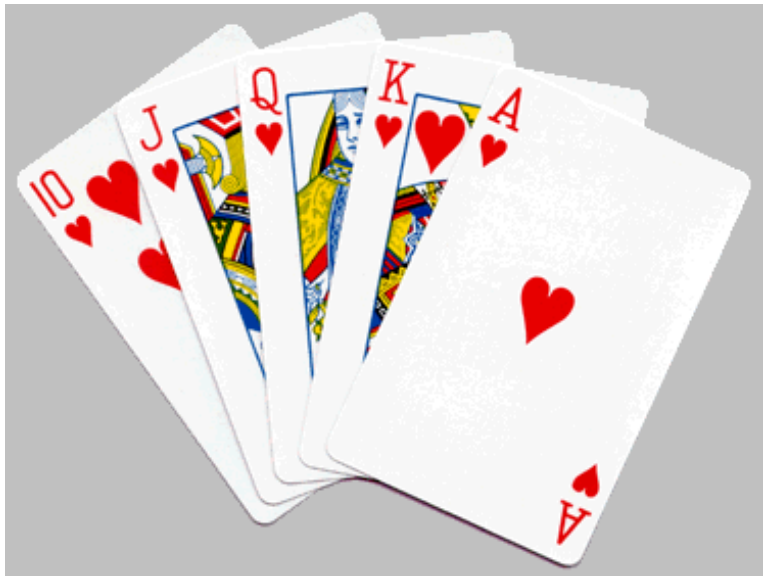
The card hands are using lots of memory. Jim therefore decides to try a more efficient representation. Whenever any two lists share a common tail, they can be updated to share one copy of the tail, and the other copy can be discarded. This process can be repeated until no two lists share a common tail.

Your job is to tell Jim how many linked list nodes he needs to store all the card hands.

## Input

The input contains several test cases followed by a line containing 0. The first line of each case contains the number of card hands. Each subsequent line describes a card hand. It starts with a number indicating the number of cards in the hand. The cards follow, separated by spaces, in the canonical order defined above. For each card, the value is given first, followed by the suit (C, D, H, or S). There are at most 100,000 cards in all hands.

## Output

For each test case, output a line containing the number of linked list nodes needed to store all the lists.

## Sample Input

```
3
3 7D AH 5S
4 9C 3D 4D 5S
2 AH 5S
0
```

## Sample Output

```
6
```