Coding test

In pari-mutuel wagering, you can place various kinds of bet. The “win” bet, for example, requires you to choose who you will think will win the race, whilst the “exacta” bet requires you to choose both first and second place horses. Say, for example, that you believe horse number 5 will win, and horse number 3 will come second, then you could create a win bet with the selection “5”, and an exacta bet with the selection “5/3”. Note here that we are using the “/” to separate the two positions – the selection “3/5” would imply that horse number 3 would win and 5 come second.

To make things more convenient for the player, we also allow combinations in our selection. So for example the win bet “3,5” is saying that I want to bet on either horse number 3 or number 5 to win. In effect, we are placing 2 bets in the one selection. Similarly, the exacta bet “2,3/7” is saying that I want two exacta bets – for “2/7” and for “3/7”. However, as it is not possible for a horse to finish both first and second, the selection “3/3” is invalid and would not count as a combination.

There are other bet types, but they all follow this general pattern. Your task is to implement functions that will count the number of combinations (i.e. discrete bets) in each selection.

**Your task is to write code to implement the “win” and “exacta” bet types, according to the interface given. We expect well-formatted code in the language of your choice with unit tests.**

type BetType interface {

  NumCombinations(selection string) int

}

Here are some examples that you can use as the basis of your tests:

|  |  |  |
| --- | --- | --- |
| Bet type | Selection | NumCombinations |
| Win | 5 | 1 |
| Win | 5,3 | 2 |
| Win | 1,3,1 | 2 |
| Win | 1/3 | 0 |
| Win | arglebargle | 0 |
| Exacta | 3/5 | 1 |
| Exacta | 3,4/7 | 2 |
| Exacta | 3/5,7 | 2 |
| Exacta | 3,4/5,7 | 4 |
| Exacta | 1,2,3/5,6 | 6 |
| Exacta | 1,2/1,2 | 2 |
| Exacta | 6/6 | 0 |

If you have any questions or queries at all please contact dtucker@i-neda.com.