Zoopla Match! Task (Backend)

This test is intended to help us gauge your competency with a language as well as how you design/structure your code. For this role we would like to see your test in the language in bold below:

- Java
- JavaScript/TypeScript
- **C#**
- Python
- Perl
- Go

Please attempt the following at your own convenience we suggest spending 90-120 minutes on it. Once you've completed the test and sent your result back, please follow it up with some notes on how you thought the test went and anything you'd do differently, given more time.

The task

Write a program to simulate a card game called "Match!" between two computer players.

"Match!" Game Rules

Game Setup

Choose a number of packs of playing cards, and combine them into a single *deck*. Shuffle the deck.

Playing the game

Cards are played sequentially from the top of the deck into the *pile*.

If two cards played sequentially *match* (see "Match condition" below), the first player to declare "Match!" takes all the cards in the pile. *For the purposes of this simulation, the program should choose a random player as having declared "Match!" first.*

Play then continues with the next card in the deck, starting a new pile. The game ends when no more cards can be drawn from the deck and no player can declare "Match!". (Any remaining cards in the pile may be discarded.)

The player that has taken the most cards is the winner. The game may end in a draw.

Match condition

The match condition determines when two cards *match* for the duration of the game. There are three options:

- The suits of two cards must match
  - Example: "3 of hearts" and "5 of hearts" match because they are both hearts.
- The values of two cards must match
  - Example: "7 of hearts" and "7 of clubs" match because they both have the value 7.
- Both suit and value must match
  - Example: "Jack of spades" only matches another "Jack of spades"

The program

As input, the program should ask:

1. how many packs of cards to use for the deck
2. which *match condition* to use

It should then simulate the game.

The program should output the results by either declaring the winner, or a draw.

Glossary

Pack - A complete set of 52 playing cards.

Deck - The set of cards in play. This could be multiple packs, or a subset of cards from a single pack depending on the game being played.

Value - The number or title value associated with a card. EG "*one*", "*queen*" or "*ace*".

Suit - A pack is divided into 4 suits. The possible suits are

- "clubs" ♣,
- "diamonds" ♦,
- "hearts" ♥, and
- "spades" ♠