

WAR!

Basic stacks and queues

Rules of the game

The card game *War* is played with a standard 52-card deck; the suits (club, hearts, diamonds, spades) play no role in this game and the ranks are ordered A, 2, 3, ..., 10, J, Q, K. Two players, Alice and Bob, each receive 26 cards. This is their initial stack.

The game proceeds in rounds as follows: Alice places her topmost card face-up in the middle of the table. Bob then places his topmost card face-up on top of it. The player with the highest card wins both cards and puts them at the bottom of his or her own stack; to be precise, the two cards are turned face-down so that Bob's card ends at the bottom.

If the two cards have equal value, they stay where they are but a "war" breaks out. During a war, each player places two cards per round instead of one, (in the order Alice–Bob–Alice–Bob). The war continues until the last two cards are not equal, whoever placed the highest last card "wins the war" and gets the entire stack. (Again, the entire stack is turned over, so Bob's last card will be at the bottom.)

Whoever runs out of cards (during a war or otherwise) loses the game.



Tips

Don't get confused by the word "stack" in the problem description. It's not necessarily a stack in the data structure sense.

When you move the cards from the middle to one of the two players after a war, you are welcome to use time proportional to the size of the war stack, even though this task *can* be performed in constant time as well.

You may use the Java Collections Framework if you want. An argument in favour is that it's good to know this library. An argument against is that you may want to implement your data structure from scratch to solidify your understanding of how the details work. A possible compromise is to use with the Collections framework until your code works, and then re-implement what you want.

Requirements

Your program reads two lines of card descriptions with obvious syntax, "As" means Ace of spades and "Tc" means Ten of clubs. The first line is Alice's, the second is Bob's, each line describes the initial stack from bottom to top.

The output either reports who won, or that the game is tied. I game that runs for more than 100,000 steps, counts as a tie.

You must produce the right output for all the *.in-files, the proper answer is in the corresponding *.out-files. Files called debug-*.in are small examples that don't use all 52 cards, these are useful for testing during early development.

You can read more about the game of War at the Wikipedia article for **War (card game)**, including variants that would make fun exercises. This exercise is strongly based on a popular entry from the ACM Programming Contest, which is described in detail in the book *Programming Challenges* by Skiena and Revilla (Springer, 2003).

Input

```
4d Ks As 4h Jh 6h Jd Qs Qh 6s 6c 2c Kc 4s Ah 3h Qd 2h 7s 9s 3c 8h Kd 7h Th Td
8d 8c 9c 7c 5d 4c Js Qc 5s Ts Jc Ad 7d Kh Tc 3s 8s 2d 2s 5h 6d Ac 5c 9h 3d 9d
```

Output

```
A wins after 2510 steps
```