Strategy

To solve the puzzle, start by moving one red circle into the pass spot. This opens a path for blue circles to cross the board. Use this opportunity to move two blue circles toward their goal. Once they've crossed, relocate the red circle from the pass spot into a corner destination.

Then, move the blue circles back to their original side to reopen the pass spot. Repeat this pattern: red enters the pass, blue crosses, red exits into a corner, blue returns. Each red circle uses the pass spot once, while blue circles move back and forth multiple times to keep the path clear.

Prioritize placing red circles into the corners first, since they're harder to reach later. Continue the cycle until all red and blue circles have swapped sides.

1. Move red into the pass spot. →
2. Move two blues across to free up the first corner. ←
3. Move red into the newly opened corner. →
4. Move two blues back to clear the pass spot. →
5. Move the next red into the pass spot. →
6. Move three blues across to free up the corner. ←
7. Move that red into the second corner. →
8. Move two blues back to open the pass again. →
9. Move the final red into the pass spot. →
10. Move two blues into their final position ←
11. Move the last red into final position →

The Key Ideas Behind This Strategy:

- **Pass control**: the pass must be controlled at all times; it's the only way to swap sides.
- **Red pieces use pass once each** to get across.
- **Blue pieces must constantly shuttle** in and out using the available empty spot created by moving red.
- **Timing matters**:
  - After you place a red, you must temporarily undo some of your blue progress to free up space.
- **You use "pushes" and "pullbacks"** to maintain flexibility.

How This Relates to Computational Thinking:

1. Decomposition

Breaking a big problem into smaller, manageable parts.

- You didn't just think "move everything" — you **separated** it:
  - Move reds through pass.
  - Move blues across in stages.
  - Manage the pass availability.
  - Handle corners early.
- Each **small action** was isolated: **first move**, **then stage**, **then cross**, etc.

---

2. Pattern Recognition

Identifying repeated structures or sequences.

- You noticed:
  - Reds always move into pass before settling.
  - Blues always shuttle back and forth.
  - The center (pass spot) is reused over and over in a predictable way.
- This **allowed you to anticipate** what steps would be needed without trial-and-error each time.

---

3. Abstraction

Focusing on important information and ignoring irrelevant details.

- You abstracted the board:
  - Instead of thinking about all 15 squares individually, you focused mainly on **corner squares**, **pass spot (1, 5, 11, 15, and 13).**
- You treated pieces not as “red” and “blue” at all times, but in terms of their **movement role** (who needs to cross, who is blocking).
- You abstracted the moves into concepts like **push**, **pull**, **stage**, and **cross**.

---

4. Algorithm Design

Creating a step-by-step procedure to solve the problem.

- You produced an **exact sequence**:
  - Move red into 13 ➔ move blues ➔ relocate red ➔ reset blues ➔ repeat.
- It's not random or guesswork — it's a **planned, repeatable** method.
- **If someone follows your procedure exactly, they will always win.**

✨ In short:

Computational Thinking Skill	How You Used It
Decomposition	Broke the puzzle into stages (corner handling, pass management, shuttling).
Pattern Recognition	Saw the repeated movement cycle around the pass spot (13).
Abstraction	Focused only on critical spaces (1, 5, 11, 15 and 13).
Algorithm Design	Wrote an explicit, generalizable procedure anyone could follow.

🚀 Why This Is Powerful:

This is exactly what computer scientists do when writing programs:

- **They simplify complex systems into clear rules.**
- **They design procedures that *always work* if followed.**
- **They manage limited resources carefully** (in this case, access to the pass spot!).

You're literally doing the mental work of a **problem solver, programmer, and system designer** — all while playing a physical game!

Fewest number of moves = 17