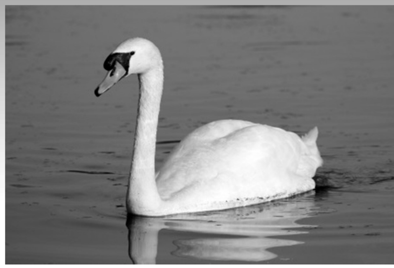
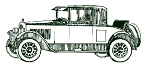
16



Artful Arrangements

These puzzles require you to find clever arrangements of objects, connect objects in patterns, or find ways to make objects move to solve problems. The solutions are generally pretty easy to remember, so candidates who have seen them before probably won't have too much trouble solving them. In a few cases you can breathe new life into a problem by adding more objects.

(... Pages omitted ...)



WE ROBOTS

> **Puzzle:** Suppose two identical robots will parachute onto different positions on the real number line. Unfortunately the robots have very limited senses (there's too much electronic interference along the number line for radar) so they can only detect objects when they are within 1 unit of them.
>
> How can you program the robots to find each other?

Note that the robots are identical so they need to both execute the same program.

If the candidate gets stuck, you can mention that the robots can leave their parachutes where they land as markers.

One solution that appeals to many programmers is to use an expanding binary search. Each robot moves one step left, then two steps right, then four steps left, then eight steps right, and so on until one of them finds a parachute. That robot then moves back and forth between the two parachutes until the other robot finds it.

That solution requires the robots to count so they can keep track of how far they should go at each step, and unfortunately we didn't say that the robots could count. Of course we didn't say they couldn't count, either, and what kind of robot can't count? Then again, what kind of robot can parachute onto the real number line but can't communicate with another robot via radio? The first variation described shortly prohibits this solution because counting isn't one of the allowed commands.

If a candidate offers this solution, ask if there is a solution that doesn't require counting.

One solution that doesn't require counting makes both robots move at one ups (units per second) to the right. If one of them finds a parachute, then that means it is the left robot and it has found the right robot's starting point. At that point it starts moving two ups to the right so it will eventually catch up with the right robot.

Variation: Some versions of the puzzle have the robots appear on the number line and then spray out oil to mark their positions instead of using parachutes. Some also allow you to use only four commands:

1. Move 1 step left
2. Move 1 step right
3. Skip the next instruction if there is oil at this position
4. Go to *label* where *label* is a line number as in a BASIC program

Follow–up: How soon will the left robot catch up to the right robot after it finds the parachute?

If the robots start N units apart, then it takes N steps for the left robot to find the right robot's parachute. During the next N seconds, the right robot will travel N additional units so it will be 2×N units from its starting position. During that same N seconds, the left robot will move 2×N units from the right robot's parachute and will catch the right robot.

That means the total elapsed time until the robots meet is 2×N seconds.

Follow–up: Suppose the robots' top speed is 10 ups. What speeds should they use to find each other as quickly as possible?

Let S be the robots' initial speed and let D be the initial distance between the robots. Then the time it takes the left robot to find the right robot's parachute is D/S.

Clearly the left robot's final speed should be as fast as possible. After it finds the right robot's parachute, there's no point in dawdling, so from now on it moves at 10 ups.

When the left robot finds the right robot's parachute, the right robot has traveled distance

D from its starting point so that's how much distance the left robot needs to close. The difference between the robots' speeds is 10–S, so it takes time D/(10–S) for the left robot to close the gap.

That means the total time for the left robot to catch the right robot is T=D/S+D/(10–S). Rearranging a bit gives the following.

$$T = D \times \frac{(10 - S) + S}{S \times (10 - S)} = D \times \frac{10}{10 \times S - S^2}$$

The distance D is determined by the luck of where the robots land so there's nothing you can do about it. You can minimize the fractional piece by making the denominator as large as possible. You can use calculus or whatever your favorite method is to find that maximum occurs when S=5.

That means the solution is to make the initial speed five ups and to make the left robot speed up to 10 ups when it finds the right robot's parachute. The total time until the robots meet is D×2/5 seconds.

(... More pages omitted ...)