

Write out `interleave`, which returns a stream that alternates between the values in `stream1` and `stream2`.

Working with an example:

Stream1 = 1, 3, 5, 7, ...

Stream2 = 2, 4, 6, 8, ...

→ result = 1, 2, 3, 4, 5, ...

Like the previous problem, you CAN just do something like:

```
(cons-stream (car stream1) (cons-stream (car stream2) (interleave cdr-stream stream1 cdr-stream stream2)))
```

But we can do better. We know that we have to return `(car stream)` the first time. So we know that some of the solution must look like this:

```
(cons-stream (car stream1) ... )
```

We want the rest `(... )` to just be a recursive call back to `interleave`. Now we know that no matter what, recursively calling `interleave` will force `(car stream1)` to be the next element. But we want to do `(car stream2)`. How do we make `stream1` be `stream2`? We flip them!

```
(cons-stream (car stream1) (interleave stream2 (cdr-stream stream1)))
```

Done!