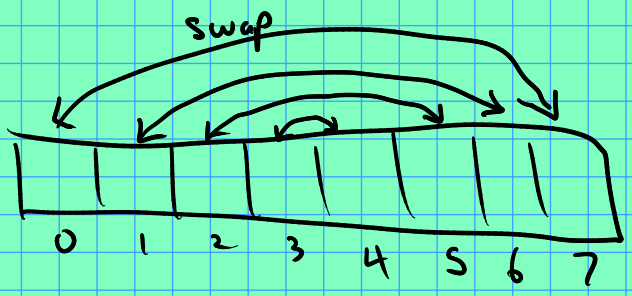
Exercise: write a function that reverses a vector. Should work "in-place" — i.e., reverse the given vector (don't return a new one).

```
void reverse(vector<int>& v);
```

Idea:



$0 \longleftrightarrow n-1$
$1 \longleftrightarrow n-2$
$\vdots$
$i \longleftrightarrow n-i-1$

← do this for $i = 0, 1, \ldots < n/2$

$n = 3$

Note on swap: the ^ operator performs bitwise exclusive OR:

$0 \oplus 0 = 0$
$0 \oplus 1 = 1$
$1 \oplus 0 = 1$
$1 \oplus 1 = 0$

(addition mod 2)

$2^2\ 2^1\ 2^0$

$7 = 1\ 1\ 1$ $\qquad$ $2 = 0\ 1\ 0$

$4+2+1 = 7$

So $7 \oplus 2 = \begin{matrix} 1 & 1 & 1 \\ \oplus & & \\ 0 & 1 & 0 \end{matrix} = 1\ 0\ 1 = 5.$

Key observation: $x \oplus x = 0$

$x \oplus 0 = 0 \oplus x = x$

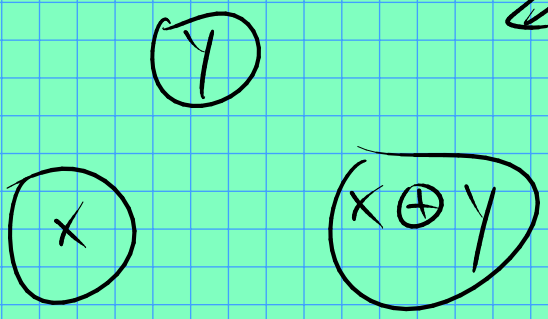
To swap x, y w/o a new variable:

```
x = x ⊕ y  // sive x "both"
y = y ⊕ x  // now y has x!
x = x ⊕ y  // places y into x!
```

(note: ⊕ y again would sive x back.)

both x

any 2 are enough to give you the 3rd.



Exercise: remove duplicates from a sorted vector, ideally performing the operation "in-place".

Warm up: allocate a new vector for the result.

Idea:
1. add V[0] to U.
2. scan V[1, 2. .] until we find a value V[i] ≠ U[0].
3. push V[i] into U.
4. "Repeat" from 2

scan V[i, i+1, ...], compare w/ U[U.size()-1]