Editing Arrays

Once we create an array, the size of that array is fixed. However, it's possible to change the values it contains.

For example, we can initialize an array that has a length of three without specifying what those values are, then later go back and edit the array to include a new value. This is useful if we know how many things we're expecting, but we don't know their specific values yet:

```
// plantHeights will be equal to [0, 0, 0]
int[] plantHeights = new int[3];

// plantHeights will now be [0, 0, 8]
plantHeights[2] = 8;
```

When we create the array with a known length but no known values, the array stores a *default type value* (0 for int, null for string). We then edit the array and swap out one of the default values with a new, specific value. In this case, we're saying that at index 2 we want to swap the default value `0` for `8`.

We can also edit the values of pre-existing arrays. Again, we can't add to the length of an existing array, but we can swap out values:

```
int[] plantHeights = { 3, 4, 6 };

// plantHeights will be [3, 5, 6]
plantHeights[1] = 5;
```

In this case, we already have an array with a defined set of values, `{ 3, 4, 6 }`. But what if a plant grows? We'll need to go back in and change its value. So if it's the second plant, we access its value using bracket notation, then change its value to `5`.

☑Instructions

1.

You've decided that you want to add Cardi B's *I Like It* to your playlist. But whoops! Only 8 tracks allowed! Swap the last song in the playlist out for *I Like It* (or another song of your choosing).

Hint

Use bracket notation to swap out a current array value for a new one:

```
string[] players = { "Emily", "Kyle", "Todd", "Rachel", "Grayson" };

// Grayson will be replaced with Chris
players[4] = "Chris";
```

2.

Since you swapped out the song, you also have to update the rating! Change the rating to reflect the playlist update.