A Reverse Fencepost

What if you want to use the same algorithm, but print the elements in reverse order? That's a little more difficult. Here is an "obvious" algorithm **which does not work correctly:**

```
cout << a[len - 1];
for (size_t i = len - 2; i >= 0; --i)
{
    cout << separator << a[i];
}
```

The loop variable type is **`size_t`**, so as soon as you print **`a[0]`** and decrement the control variable **`i`**, instead of becoming **`-1`**, it "wraps around" and becomes the **largest possible unsigned number**. Since array subscripts are **not range checked**, the loop prints at larger and larger indexes until the program crashes.

This example below **works correctly**. Notice the extra **`if`** statement:

```
if (len > 0)
{
    cout << a[len - 1];
    for (size_t i = len - 1; i > 0; --i)
    {
        cout << separator << a[i - 1];
    }
}
```