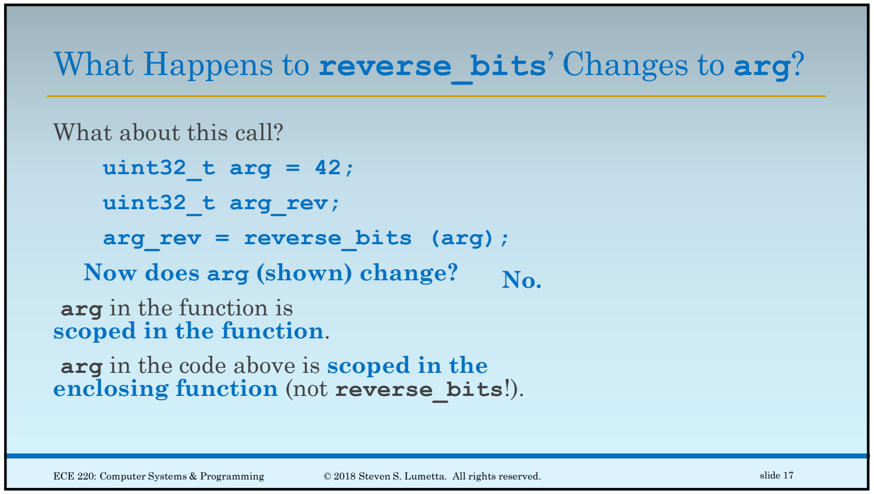


## What Happens to **`reverse_bits`**' Changes to **`arg`**?

What about this call?

```
uint32_t arg = 42;
uint32_t arg_rev;
arg_rev = reverse_bits (arg);
```

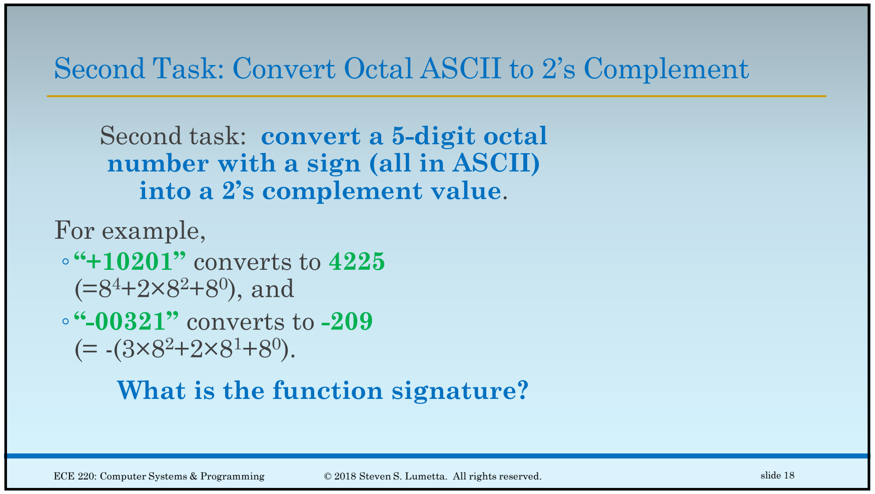
**Now does `arg` (shown) change?** **No.**

`arg` in the function is **scoped in the function**.

`arg` in the code above is **scoped in the enclosing function** (not `reverse_bits`!).

ECE 220: Computer Systems & Programming © 2018 Steven S. Lumetta. All rights reserved. slide 17

17



## Second Task: Convert Octal ASCII to 2's Complement

Second task: **convert a 5-digit octal number with a sign (all in ASCII) into a 2's complement value.**

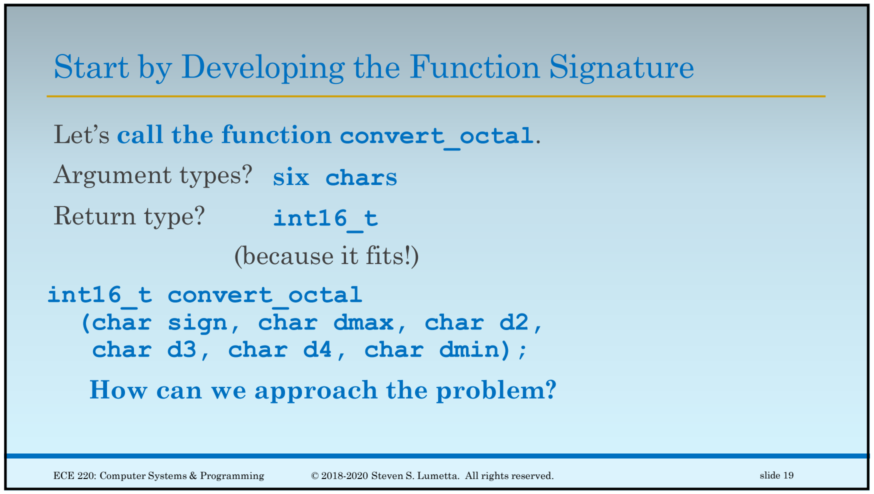
For example,

- **"+10201"** converts to **4225** ($=8^4+2\times8^2+8^0$), and
- **"-00321"** converts to **-209** ($= -(3\times8^2+2\times8^1+8^0)$).

**What is the function signature?**

ECE 220: Computer Systems & Programming © 2018 Steven S. Lumetta. All rights reserved. slide 18

18



## Start by Developing the Function Signature

Let's **call the function `convert_octal`**.

Argument types? **`six chars`**

Return type? **`int16_t`**

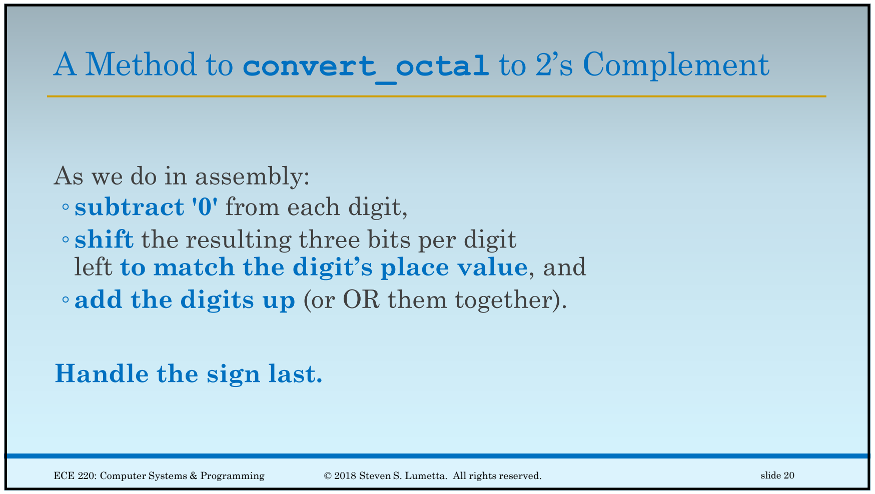
(because it fits!)

```
int16_t convert_octal
  (char sign, char dmax, char d2,
   char d3, char d4, char dmin);
```

**How can we approach the problem?**

ECE 220: Computer Systems & Programming © 2018-2020 Steven S. Lumetta. All rights reserved. slide 19

19



## A Method to **`convert_octal`** to 2's Complement

As we do in assembly:

- **subtract '0'** from each digit,
- **shift** the resulting three bits per digit left **to match the digit's place value**, and
- **add the digits up** (or OR them together).

**Handle the sign last.**

ECE 220: Computer Systems & Programming © 2018 Steven S. Lumetta. All rights reserved. slide 20

20