Finish the Assignment Operator

The right side's value is now in **R0**.

```
abs_value = (0 <= num ? num : -num);
```

Let's store it into **abs_value**.

Where is abs_value?

Look in the symbol table!

scope	identifier	type	from	offset	...
translate.c	the_number	int32_t	R4	0	
find_abs	abs_value	int32_t	R5	0	...
find_abs	num	int32_t	R5	4	...

Store R0 into Variable abs_value

```
LDR R0,R5,#4
BRn ELSE_COND
LDR R0,R5,#4
BRnzp DONE_COND
ELSE_COND
LDR R0,R5,#4
NOT R0,R0
ADD R0,R0,#1
DONE_COND
STR R0,R5,#0
```

Store R0 into **abs_value**.

Is there an LC-3 instruction for that?

identifier	type	from	offset
the_number	int32_t	R4	0
abs_value	int32_t	R5	0
num	int32_t	R5	4

We Have Translated the First C Statement!

```
LDR R0,R5,#4
BRn ELSE_COND
LDR R0,R5,#4
BRnzp DONE_COND
ELSE_COND
LDR R0,R5,#4
NOT R0,R0
ADD R0,R0,#1
DONE_COND
STR R0,R5,#0
```

The statement is complete!

```
abs_value =
    (0 <= num ?
     num : -num);
```


Implement the Second (and Last) C Statement

Here's the second statement.

```
return abs_value;
```

(Copy **abs_value** to return value, then RET.)

Where is abs_value?

Look in the symbol table!

scope	identifier	type	from	offset	...
translate.c	the_number	int32_t	R4	0	
find_abs	abs_value	int32_t	R5	0	...
find_abs	num	int32_t	R5	4	...

