Set Frame Pointer for `find_abs`

```
FIND_ABS
ADD R6,R6,#-4
STR R5,R6,#1
ADD R5,R6,#0
```

Next, set R5 to point to the lowest local variable.

Is there an LC-3 instruction for that?

How much do we add?

Note: amount added depends on space for local variables.

Save Return Address into the Stack Frame

```
FIND_ABS
ADD R6,R6,#-4
STR R5,R6,#1
ADD R5,R6,#0
STR R7,R5,#2
```

Finally, save R7 into the stack frame.

Is there an LC-3 instruction for that?

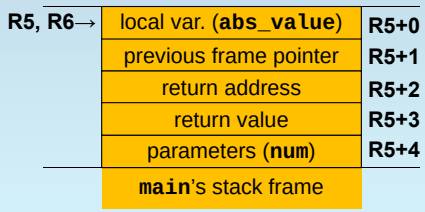
What are the base register and offset?

Note: always the same offset from R5.

Stack Frame for `find_abs` (During Execution of Code)

Now we can write code for the C statements.

Note that offsets match the symbol table.



Implement the First C Statement

Here's the first statement.

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

We start with the **test**.

Where is num?

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	...

