

## Use a `char*` to Point to Stack Element to be Written

Decrement, then use index.

Copy to this element on stack.

```
write = s->data[--s->top];
```

## Loop Until End of String or Out of Space

Loop until end of str.

```
for (i = 0; '\0' != *str; i++) {
    if (199 == i) {
        s->top++;
        return 0;
    }
    *write++ = *str++;
}
```

If `str` is too long, restore stack top and fail.

Copy a character and advance pointers.

## End the String and Return Success

Write NUL to end of string.

```
*write = '\0';
return 1;
}
```

Push has succeeded.

## Must Also Copy in `stack_pop`

### What about popping a string?

We make a copy in `stack_push`.

After `stack_pop` returns,

- the copy is no longer on the stack,
- thus a call to `stack_push` will overwrite it
- so we should not return a pointer to the copy.

### The implication?

`stack_pop` must also make a copy.