Comparing C-Strings

Do not use the relational operators (`<`, `==`, etc.) to compare C-strings. Instead, use the library function `strcmp()`, which compares `s1` and `s2` **lexicographically** and returns an integer indicating their relationship:

- **Zero** if the two strings are equal.
- **Negative** if the first string lexicographically precedes the second string. (Lexicographically simply means "in dictionary order").
- **Positive** if the first string lexicographically follows the second string.

To use `strcmp()` correctly:

- Call the function and save the int it returns.
- Use the returned value with a relational operator.
- **Don't** treat the return value from `strcmp()` as a Boolean expression.
- Don't repeatedly call `strcmp()` on the same strings (inefficient).

Here's a quick example. The C-strings `s1` and `s2` are initialized elsewhere. Since we don't need to modify either argument, we can use "pointer-style" C-strings.

```
const char *s1 = ..., *s2 = ...;

int result = strcmp(s1, s2);

if (result == 0) ...             // equal
else if (result < 0) ...         // s1 < s2
else ...                         // s1 > s2
```