

String basics

When you think about data, various images may come to mind – text, sequences of numbers, or even a combination of different types. There's a versatile data type that can represent this wide variety of data, and that's the **string**.

A string is an ordered array of characters. The character can be anything: a letter of the Greek alphabet, a number, or a strange Unicode symbol ☞.

As mentioned earlier, strings are an **ordered array**. This means that every character in a string corresponds to an index. The counting of characters in strings traditionally starts from zero.

A string can have zero length. In this case, we call it an **empty string**.

We can perform numerous operations on strings. Let's take a brief look at a few of them:

- `reverse(s)` returns the reversed string, i.e. the string written backward. For example, `reverse(LIVE) = EVIL`;
- `concat(s,t)`
- `compare(s, t)`
- `get_symbol(s, i)` returns the character in the given string at given index
- `length(s)`

Obviously, the only nonproper substring of a string is the string itself. An empty string is a substring of any string.

The whole word is a suffix and a prefix for itself as well. Also, empty prefixes and suffixes can exist.

What if a certain prefix and suffix coincide? We end up with a new term: a **border**. Formally, the border of a string is a non-empty substring that is both a proper prefix and a proper suffix of the string.

The longest border of **ATTA** is **A**.

Of course, a string may not have a border, because not always there is a proper prefix that is equal to a proper suffix. A string is called **unbordered** if it's impossible to find any border in it.