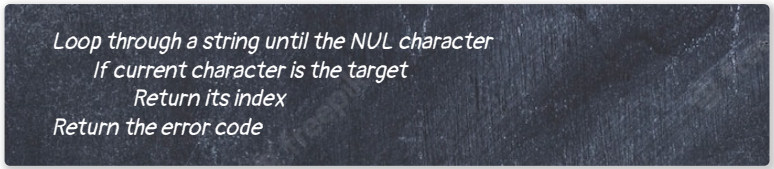
Find First

To find the first occurrence of a particular character in a string, you'd employ the **linear search** algorithm:



Assuming that we use `-1` for the error code then an **array-notation** implementation of the function could look like this:

```
int find(const char a[], char target)
{
    for (int i = 0; a[i] != '\0'; ++i)
        if (a[i] == target)
            return i;
    return -1;
}
```

A (more cryptic) **pointer-notation** implementation might look like this:

```
int find(const char* s, char target)
{
    auto *p = s;
    while (*p && *p != target) p++;
    if (*p) return p - s;
    return -1;
}
```

The **temporary pointer** `p` is moved through the C-string `s`. The expression `*p` is false when the `NUL` is encountered. Since the loop **must end** when you encounter the `NUL`, or, when you find the `target`, you know that the loop **terminates in every case**.

After the loop is over, there are **two** possibilities. If `p` is pointing at **any** character, it **must** be the `target` character. That means you can use **pointer difference** to return the index. Otherwise, `p` **must** be pointing at the `NUL` character and you can return `-1`.