Practice It

••• 1. Suppose you are given the task of printing all characters in a string `str` that occur exactly once. For example, given the string `"reader"`, you would print `a` and `d` because `r` and `e` are repeated.

That is not easy, so let's solve a simpler problem first. Does the *first* character occur exactly once?

Start with the algorithm from Section 4.7.3. You need to modify the algorithm so that you look for the first character, and you need to look for it in the *rest* of the string.

Print the first character when it doesn't occur in the rest of the string.

Rearrange the following lines of code to produce a solution. Not all lines are useful.

Order the statements by dragging them into the left window. Use the guidelines for proper indenting.

✓GOOD JOB!

```
bool found = false;
string target = str.substr(0, 1);
int position = 1;
while (!found && position < str.length())
{
    string ch = str.substr(position, 1);
    if (ch == target) { found = true; }
    else { position++; }
}
if (!found) { cout << target << endl; }
```

```
if (found) { cout << ch << endl; }
if (ch == "A" || ch == "a") { found = true; }
int position = 0;
if (found) { cout << target << endl; }
```

8 correct, one error, 38 seconds

Start over

• • • 2. In the preceding problem, you determined whether the first character of a string `str` occurs exactly once in the string. Now let us solve a slightly harder problem. Does the i^{th} character occur exactly once?

Adapt your solution to deal with this case. Note that you need to skip position `i`.

Rearrange the following lines of code to produce a solution. Not all lines are useful.

Order the statements by dragging them into the left window. Use the guidelines for proper indenting.

✓GOOD JOB!

```
bool found = false;
string target = str.substr(i, 1);
int position = 0;
string ch = "?";
while (!found && position < str.length())
{
    ch = str.substr(position, 1);
    if (ch == target && position != i) { found = true; }
    else { position++; }
}
if (!found) { cout << target << endl; }
```

```
if (found) { cout << target << endl; }
int position = 1;
string target = str.substr(0, 1);
if (ch == target) { found = true; }
if (found) { cout << ch << endl; }
if (ch == "A" || ch == "a") { found = true; }
```

••• 3. In the preceding problem, you determined whether the i^{th} character of a string `str` occurs exactly once in the string. Now you are ready to solve the original problem, to print all characters that occur exactly once.

Rearrange the following lines of code to produce a solution. Because `ch` is not needed outside the `while` loop, declare it inside.

Order the statements by dragging them into the left window. Use the guidelines for proper indenting.

✓GOOD JOB!

```
for (int i = 0; i < str.length(); i++)
{
    bool found = false;
    int position = 0;
    string target = str.substr(i, 1);
    while (!found && position < str.length())
    {
        string ch = str.substr(position, 1);
        if (ch == target && position != i) { found = true; }
        else { position++; }
    }
    if (!found) { cout << target << endl; }
}
```