

Lab 4.5.8 Text manipulation: templates

Objectives

Familiarize the student with:

- the idea of a text template and template filling.

Scenario

When running a newsletter or preparing announcements targeted at a group of people, one might need to prepare many very similar messages, differing only in minor details.

Write a program that reads two lines of text.

The first line should contain a list of name value pairs separated by commas. Each pair should be in the following form: "name=value".

The second line should contain a template that should be filled with the values from the first line. A placeholder should have the form "[name]" and should be replaced by an appropriate value.

If your program encounters a placeholder with an unknown name, the placeholder should be left unprocessed in the output text.

```
#include <string>
#include <iostream>

int main()
{
    std::string values;
    std::getline(std::cin, values);

    std::string template;
    std::getline(std::cin, template);

    // Put values into the template

    std::cout << template << "\n";
}
```

Example input

```
person=Paul,action=strolls around,place=the park
[person] [action] [place]
```

Example output

```
Paul strolls around the park
```

Example input

```
customer=John,pet=dog
Dear [customer], we were happy to take care of your [pet] while you were away. Please call our pet hotel again if you need help with your [pet]!
```

Example output

```
Dear John, we were happy to take care of your dog while you were away. Please call our pet hotel again if you need help with your dog!
```

Example input

```
date=02/02/02 3:00PM,room=504
Dear [customer], your exam will take place on [date] in room [room]
```

Example output

```
Dear [customer], your exam will take place on 02/02/02 3:00PM in room 504
```