

CPSC 535 Advanced Algorithms

Project 1: Rewriting greeting cards

Prof. Doina Bein, CSU Fullerton

dbein@fullerton.edu

Introduction

In this project you will design and implement one algorithm related to strings. You will design the algorithm, describe the algorithm using clear pseudocode, and implement it using C/C++/C#/Java/Python, compile, test it, and submit BOTH the report (as a PDF file) and the files. The execution should take less than one hour.

Rewriting greeting cards

People appreciate small gestures like receiving a greeting card on special occasions, even if it is delivered electronically. Sometimes writing this greeting takes longer than planned because it is not easy to come up with something thoughtful. But once you do it, why not reusing most of the text for another occasion? For example, a congratulation email on a new car may be similar to a congratulation email on a new house. One can copy and paste parts of the previous greeting to make it into a new greeting, but we would like to develop software that does it for us. The problem can be formulated as follows: Given a string S of length $N > 0$, and a list of M pairs of strings LS , each pair representing a string to be replaced and the second being the actual replacement, display the new string R that is obtained by replacing every occurrence of the first string in each pair with the corresponding second string in the pair. The string matching must be case sensitive.

Example 1:

Input:

$S[] =$ "My dear Anna, let me congratulate you on the beautiful car that you purchased today. It looks very posh. I hope you got a good deal. Cars are expensive but much needed. Best regards, Naomi."

$N = 190$

$M = 4$

$LS[] = \{ \{ \text{"Anna"}, \text{"Jovi and Victor"} \}, \{ \text{"car"}, \text{"house"} \}, \{ \text{"today"}, \text{"last week"} \}, \{ \text{"posh"}, \text{"well built"} \} \}$

Output:

$R[] =$ "My dear Jovi and Victor, let me congratulate you on the beautiful house that you purchased last week. It looks very well built. I hope you got a good deal. Cars are expensive but much needed. Best regards, Naomi."

Example 2:

Input:

$S[] =$ "Our newest students have been asked to stay today until the end of the classes. The old principal."

$N = 99$

$M = 5$

```
LS[] = { {"new", "old"}, {"student", "teacher"}, {"to", "yester"}, {"old", "young"}, {"end", "beginning"} }
```

Output:

R[] = “Our oldest teachers have been asked yester stay yesterday until the beginning of the classes. The young principal.”

Example 3:

Input:

S[] = “Our newest students have been asked to stay today until the end of the classes. The old principal.”

N = 99

M = 5

```
LS[] = { {"old", "young"}, {"student", "teacher"}, {"to", "yester"}, {"end", "beginning"}, {"new", "old"} }
```

Output:

R[] = “Our oldest teachers have been asked yester stay yesterday until the beginning of the classes. The young principal.”

Variables N and M can be computed from S[] and LS[] so it is up to you if you want the user to input them or not. The variables S[] and LS[] must be read from the user, most preferred from a file.

The output should display the resulting string R. Please check that your program executes correctly on the two given examples. Please create more examples on your own and test them.

Grading rubric

The total grade is 35 points. Your grade will be comprised of three parts, Form, Function, and Report:

- Function refers to whether your code works properly (18 points).
- Form refers to the design, organization, and presentation of your code. The instructor will read your code and evaluate these aspects of your submission (6 points):
 - README.md completed clearly = 2 points
 - Style (whitespace, variable names, comments, helper functions, etc.) = 2 points
 - **Language C++** Craftsmanship (appropriate handling of encapsulation, memory management, avoids gross inefficiency and taboo coding practices, etc.) = 2 points
- Report (11 points) divided as follows:
 - Summary of report document (2 points)
 - Pseudocode of the algorithm (6 points)
 - Four screenshots: one for the group members and three for the three sample input files (1 point each, total 3 points)

Obtaining and Submitting Code

This document explains how to obtain and submit your work:

[GitHub Education / Tuffix Instructions](#)

Here is the invitation link for this project:

<https://classroom.github.com/g/74HxdnCq>

Implementation

You are provided with the following files.

1. README.md contains a brief description of the project, and a place to write the names and CSUF email addresses of the group members. You need to modify this file to identify your group members.
2. LICENSE contains a description of the MIT license.

What to Do

First, add your group members' names to README.md. Then write clear pseudocode for the rewriting algorithm, describe what parameters are needed to execute your program, and submit it as a PDF report. Your report should include the following:

1. Your names, CSUF-supplied email address(es), and an indication that the submission is for project 1.
2. A full-screen screenshot with your group member names shown clearly. One way to make your names appear in Atom is to simply open your README.md.
3. The pseudocode for the algorithms
4. A brief description on how to run the code.
5. Three snapshots of code executing for the three given examples.

Then implement your algorithm in C/C++/C#/Java/Python. Submit your PDF by committing it to your GitHub repository along with your code.