

COSC2436 hw1: String Manipulation and Filtering

1. Introduction

You will implement a C++ program to decode different string of number from the input file then filtering out the information requested by the command file. This assignment focusses on array management and string handling in C++. You will also learn about string manipulation with predefined string utility functions and basic file operations.

2. Input files

- The input file will contain a list of encoded id number. The input can range anywhere from 0 to 100 entries. Input might contain empty lines and spaces (empty lines and spaces need to be removed to process each entry correctly).
- Each entry will be on its own line and have 2 different parts, coded characters set and id string.
- The id string will consist of digits, alphabet characters, and the hash sign.
- Alphabet characters in the id string will always have a number representation for it in the characters set, separated by a colon.
- Semicolon is used to separate characters in the set and id string.
- The id string will always come after the characters set in the entry.
- Valid entry should have all 2 parts, the characters set and the id string.
- Invalid entry should be ignored.

- Example of valid entry:

a:123;b:456;c:789;id:c11ba3#2b#a

*characters set in yellow, id string in green

- Example of invalid entry:

a:123;b:456;c:789 – missing id string

id:c11ba3#2b#a – missing characters set

3. Decoding process

Example input: **a:123;b:456;c:789;id:c11ba3#2b#a**

- Replace all the alphabet characters in the id string with its number representation in the characters set.

a:123;b:456;c:789;id:c11ba3#2b#a

⇒ **789114561233#2456#123**

- The hash sign should be replaced by its index in the id string (after replacing all the alphabet). The first hash sign is at index 12, and the second hash sign is at index 17

789114561233#2456#123

12 17

⇒ **78911456123312245617123**

4. Command files

- The command file will contain different criteria for filtering out id.
- There will be a total of two types (case sensitive): “first4” and “last4”.
 - first4: those id that have its first 4 digits match with the value given in the command.
 - last4: those id that have its last 4 digits match with the value given in the command.
- The criteria and its value will be separated by a colon. Ex: “first4:1234”, “last4:6789”.
- If multiple value appears for the same criteria in the command file, then as long as the id satisfy one of those value then it should be printed to the output.
 - Ex: “first4:1234”, “first4:6789” => id that have 1234 or 6789 as the first 4 digits should be printed to the output.
- If both criteria appear in the command file, then the id must satisfy with one of the values of each criteria in order to be in the output.
 - Ex: “first4:1234”, “first4:6789”, “last4:1357”, “last4:2468” => id that have 1234 or 6789 as the first 4 digits and 1357 or 2468 as the last 4 digits should be printed to the output.
- Each value in the command will be on its own line, there might be empty lines and spaces.
- If the command is empty, all the valid id should be printed to the output.

5. Output files

- The output file should contain all the valid id filtered out by the command file in the order they appeared in the input file (empty lines and spaced should be removed).
- Each entry should be on its own line.
- If there are no id that satisfy the criteria in the command file, then the output file should be empty.

6. Requirements

Homework is individual. Your homework will be automatically screened for code plagiarism against code from the other students and code from external sources. Code that is copied from another student (for instance, renaming variables, changing for and while loops, changing indentation, etc. will be treated as copy) will be detected and result in "0" in this homework. The limit is 50% similarity.

7. Turn in your homework

Homework 1 needs to be turned in to our Linux server, follow the link here

https://rizk.netlify.app/courses/cosc2430/2_resources/

Make sure to create a folder under your root directory, name it "hw1" (case sensitive), copy all your .cpp and .h file to this folder, "ArgumentManager.h" need to be included as well.

PS: This document may have typos, if you think something illogical, please email TAs for confirmation.