ATILIM UNIVERSITY

DEPARTMENT OF COMPUTER ENGINEERING

Computer Programming 2 (CMPE114)

Assignment-3

2020-2021 Spring Semester

Instructors

Asst. Prof. Dr. Cansu Çiğdem EKİN

Asst. Prof. Dr. Gonca Gökçe MENEKŞE DALVEREN

Prepeared by

Res. Asst. Ege Bulut UYGUR

Due date: 19/05/2021

RULES & REGULATIONS

- You must submit your homework via Moodle before the due date/time.
- Homework submitted via email will be ignored.
- Late submissions will NOT be graded.
- Before submission check that:
    1) Use CodeBlocks or Dev-C++ as the IDE
    2) Name your homework file as in the format lastname_firstname_hw1.c
    3) Do NOT use Turkish characters when you name C source file.
    4) Your homework must be a source file (a C file not CPP).
    5) Do NOT upload .exe file. Otherwise you will get ZERO.
- Your codes will be checked by special software (JPLAG & MOSS) for code similarity. If the code similarity between any two or more submissions is higher than %90, we will also examine and compare these codes by eye. If we are convinced that the similarity between two codes is not merely a coincidence, all involved HWs will get 0 as the grade.
- Therefore;
    1) Group study is not allowed. Everyone needs to do his/her homework as an individual.
    2) Do not use Internet resources! If any other student use the same source then your submissions will be found as similar!
    3) Do not share your solution/ideas with others
- Moreover, if we believe that the students cheat on these assignments we disciplinary act will be initiated.
- GRADING POLICY: If your code can not run at all or as expected, your code will be evaluated over 70 points not 100 points.

Assignment

Write a program that fixes the mistakes within the given strings of input paragraphs. Firstly, the program will take an input string from the user. Your string can be of any size but there should be at least hundred characters available. The program will check three rules that are given to you below and apply them over the input string that the user enters. These rules will be checked by the functions your program holds in order. Therefore they will be applied over the input(s) in order as rule 1 and rule 2. The information for the rules are given below.

Rule 1 : There must not be any duplicate letters for the inputs that the user enter. If a duplicate letter or space character is detected, all of the extra letters should be deleted until only one of the letters remain. This condition of being 'duplicate' and/or 'multiple' is constrained by some rules. The rule is applied over only consecutive character forms that are right next to each other.

Example

Base Version: I wass waalkinng in thhee Ankarraa.

Fixed Version: I was walking in the Ankara.

You are expected to apply this rule inside a function. Function will have a void return type and will accept the input string from the main as parameter. Your function will be called from the main, fix the string and update it to it's correct form by regarding to this rule.

Rule 2 : Eliminate unnecessary space characters, add space characters if you need to. The spacing structure rule for the given paragraphs is given as;

1- Punctuation marks must not have any space between them and the word before them (the word they belong to). Any punctuation must be the first character to come right after a word is finished.
2- After every word with or punctuation mark, there must always be a single space character before the next word starts. (There is no need to apply this at the end of the input string). This means, if there are no space between a punctuation and the next word's first character, you must add a space character between the punctuation and the beginning of the next word.

After applying rule 2, you will call the rule 2's function from the main and perform the space elimination, addition over the string or just stay neutral since there might be no need for fixing spaces after the appliance of rule 1 over certain strings. The function should apply the rules given.

Sample Paragraphs:

- Paragraph 1: Ppeaace is the resullt of retrraining yYyyour miiind to proccess liife as it is ,ratheer than as you think it shhould be .
- Paragraph 2: ssSpaccce TttrRavel is life - enhancing,, AaaAnd anything that' s life - enhanncing is woOorth doinnng….

Expected Output:

- Fixed Paragraph 1: Peace is the result of retraining your mind to proces life as it is, rather than as you think it should be.
- Fixed Paragraph 2: space Travel is life- enhancing, and anything that's life-enhancing is worth doing.

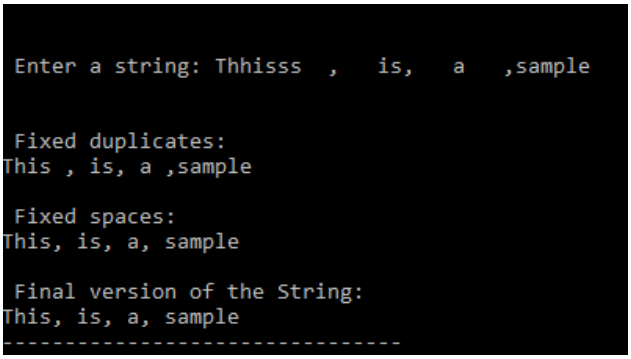
After the program ends, you will output the final version of the string. In between the appliance of these rules, you will print the result of the applied rules as output. Sample runs and important remarks are given in the documentation further below.

Important Remarks

- You can code your assignment with or without the help of the string.h and/or ctype.h libraries functions. Algorithms are up to you as long as you can perform the necessary string operations and build the correct coding structure as expected.
- You can define any variables of any type you need as long as the general program structure is working as the given guidelines in your assignment document.

Sample Outputs

Output 1:



Output 2: