Lab 2: Anagram Checker

Professor Patt found that anagrams are very interesting. Anagrams are words or phrases that have the same characters but in a different order. For example, "listen" and "silent" are anagrams, "eleven plus two" and "twelve plus one" are anagrams, "dormitory" and "dirty room" are also anagrams. As a computer man, Professor Patt would like to let computer do the check. Your job is to write a program to check if two given strings are anagrams.

Implementation Details

- You are required to write in **LC-3 assembly language**.
- Your program should start at x3000.
- The two given strings are stored in memory, and their start addresses are placed at x4000 and x4001. You are only required to compare characters `a-z` . Here comparation is case-insensitive, which means "Listen" and "Silent" are anagrams. Blanks should be ignored, which means "dormitory" and "dirty room" are anagrams. Other characters will not appear in the given strings.
- After check, output `YES` or `NO` to the console as the result. Use instruction `TRAP x21` (OUT) to output a char or use instrucion `TRAP x22` (PUTS) to output a string.
- You can write code to load data in x4000 like this:

  ```
            .ORIG x4000
            .FILL str1
            .FILL str2
  str1      .STRINGZ "listen"
  str2      .STRINGZ "silent"
            .END
  ```

  Actually you can write more than one `.ORIG` and `.END` pairs in a single .asm file, in order to making your code distributed in different memory addresses.
- Remember to halt your program after execution.

Limitations

- the length of one given string: $0 < l \leq 100$

Grading

Lab 2 takes **5%** of the total score, consisting of Check part (50%) and Report part (50%).

Check Part

- First upload your code to Learning in ZJU, then find a TA to check your code in person. TAs will first test the correctness of your program, then ask you some questions to make sure you understand what you code but not cheat.
- You can try again if you fail in checking, but there will be a penalty of -10% (of checking part) for each try.
- We strongly suggest you to make a thorough test by yourself before checking.

- We strongly suggest you to write enough comments in your code so that you will be aware of what's going on in your program and confident to answer TA's questions.

Report Part

- Report must be written **in English**, concise and carrying main ideas. Try to use the report to convince TAs that you complete the task by yourself.
- Your lab report should contains the following contents:
    - Algorithm. Flowchart or Pseudocode is prefered. The complexity of your algorithm will not affect your score.
    - Essential parts of your code with sufficient comments. Please only select the most important code phases and explain them.
    - Questions that TA asked you, and Answers.
- **No more than 2 A4 pages.** No template provided. Be sure to make it readable.

Penalty

- **Wrong Answer**: -10% of Check part each time.
- **Delay**: -20% of the corresponding part per day.
- **Cheating**: -100% of this lab. Additionly, -10% of the final score of this course. **Please note that uploading your answer to the Internet is also CHEATING!!!**