Middle East Technical University

Department of Computer Engineering

CENG 229

Fall 2019

Preliminary for Lab Exam 3

REGULATIONS

This is a preliminary question for Lab Exam 3.

Submission: No submission

Team: There is **no** teaming up. Try to solve the given question on your own.

PROBLEM

In this preliminary question and the lab exam, you will practice **Pointers**, **Arrays**, and **Functions** in C programming. You will implement specified functions that can be used in text editors. In your implementation, you are **not** allowed to use the `string.h` library and its functions.

Your functions will work on a one-dimensional character array which includes the text. This array is defined globally as `char text[SIZE];` where `SIZE` indicates the size of the array. The size of the array and its content are provided by us.

The text contains alpha-numeric characters, punctuation, and the space character. There is only one end-of-line character (`\n`) at the end of the text content. The characters used in the text are listed in Table 1.

The content of the `text` array is composed of words. Words are separated by **at least one** space character. The `text` array can contain characters of the Table 1.

Table 1: Non-space characters of the text array content.

Uppercase Letters	`A...Z`
Lowercase Letters	`a...z`
Numbers	`0...9`
Dot	`.`
Coma	`,`
Exclamation mark	`!`
Question mark	`?`
Apostrophe	`'`
Dash	`-`

SPECIFICATIONS

- Write a function, `void countCharacters(int *charCountP)`, that counts **non-space characters** in the text array and puts the result into the variable pointed by `charCountP`.
- Write a function, `void countWords(int *wordCountP)`, that counts **the number of words** in the text array and put the result into the variable pointed by `wordCountP`. Please note that, words can be separated by more than one space characters.
- Write a function, `int searchWord(char *searchedWordP, int *startP, int *endP)`, that searches for a given word whose pointer is provided in `searchedWordP`. If the word is found; the function,
  - puts the starting position of the word into the variable pointed by `startP`,
  - puts the ending position of the word into the variable pointed by `endP`,
  - returns **1**.

  If the word is not found, the function returns **0**. If there are more than one occurrences of the searched word in the text, it just provides results for the first occurrence, then stops searching.
- Write a function, `void removeSpaces()`, that recognizes if there are **more than one** space characters and removes the unnecessary ones, decreasing the number of spaces to **one**. Please note that, after this action you should align the text to the left. For example if the content of the text array is
  `I  have a  dream!  \n`
  it must be converted to
  `I have a dream!  \n`
  after the function call.
- The text array is created and initialized by us. Also, there is a global variable called `word` used for holding the word (as a C-string) to be searched. The `void printText()` function, that prints the `text` array, is also provided by us.
- You can assume the punctuation is a character of the word. For example, the word `we'll` is a 5 character word including the apostrophe.

EXAMPLE RUN

Assume that the text array is initialized as follows:

```
My          message is that we'll be      watching you. \n
```

Example function `runLab3()` is provided as follows:

```
void runLab3()
{
int wordCount=0, characterCount=0, startPosition, endPosition;
scanf("%s", word); /* word is a global variable */
printText();
countCharacters(&characterCount);
countWords(&wordCount);
printf("%d\n", characterCount);
printf("%d\n", wordCount);
if(searchWord(word, &startPosition, &endPosition))
printf("%s %d %d\n", word, startPosition, endPosition);
else
printf("No result\n");
removeSpaces();
printText();
}
```

After providing the input as `we'll` the output is acquired as follows:

```
My          message is that we'll be      watching you.
34
8
we'll 28 32
My message is that we'll be watching you.
```