CS201 – Spring 2022-2023 - Sabancı University

Homework #3: A Simple Search Engine

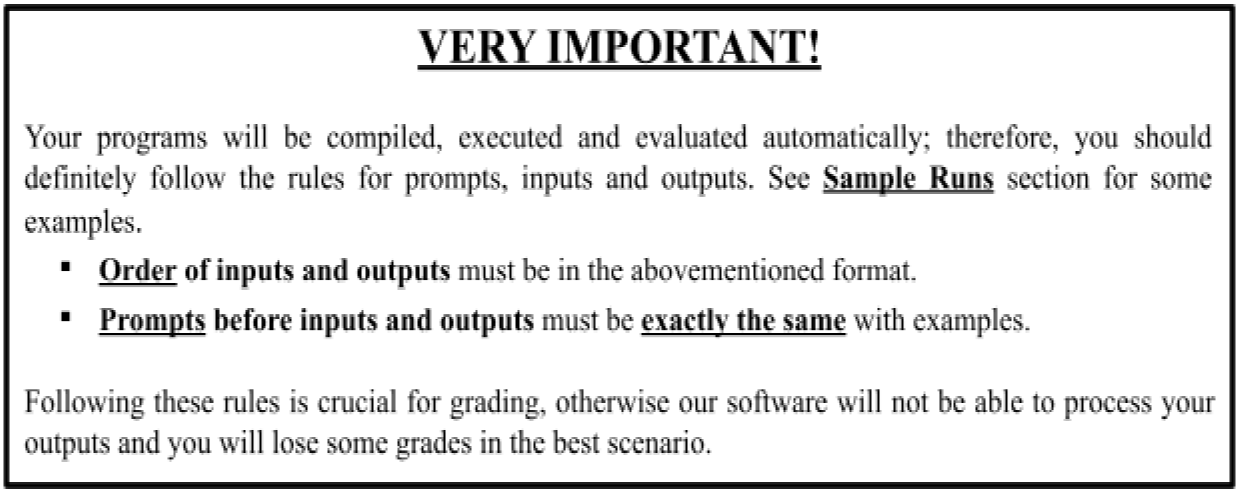
Due April 19, Wednesday, 23:00 (Sharp Deadline)

Introduction

In this homework, you will write a simple program that searches for a string in another string. There are some special searching parameters and details about it will be given in later sections. First, your program will get a string until the word "end" (case-insensitive) is present and then it will ask for a search string. Your program should be able to do multiple searches until the search string "quit" (case-insensitive) is entered. The aim of this homework is to practice on string member functions and loops. Of course, you will need to use the topics that we have seen before such as if-else statements and functions, as needed.

Your homework will be automatically graded using SUCourse, so it is very important to satisfy the exact same outputs given in the example test cases of SUCourse. Please submit your assignment by writing your main source (cpp) file content into the Answer field. You can utilize the **Check** button under the code editor at SUCourse to check whether your implementation is working in the expected way. After you check your solution code, you will see your grade with the example test cases used; however your homework will then be graded with **different** test cases.

To submit your homework, you must hit the **"Finish attempt…"** and **"Submit all and finish"** buttons. **Just a reminder of a character ↩ which refers to a newline in your expected output.**



VERY IMPORTANT!

Your programs will be compiled, executed and evaluated automatically; therefore, you should definitely follow the rules for prompts, inputs and outputs. See **Sample Runs** section for some examples.

- **Order of inputs and outputs** must be in the abovementioned format.
- **Prompts before inputs and outputs** must be **exactly the same** with examples.

Following these rules is crucial for grading, otherwise our software will not be able to process your outputs and you will lose some grades in the best scenario.

Inputs

There are two inputs respectively; Source String and Search String. Both inputs are strings. You should note that there can be spaces between the words in the Source String (hint: use a while loop to read all words using cin). However, the source string cannot contain any characters other than numbers, letters and space in the Source String, such as punctuation. On the other hand, you can assume that there is no space or tab in the search string, a single word will be entered, and you do not need to check the input for this case. Search String must end with one of ‘+’ , ‘.’ , ‘*’,’**’ characters. (See Sample Runs). Your program should get the Source String only once, but ask multiple Search Strings until “quit” (case-insensitive) is entered.

Input Checks

All inputs must be checked. In the case of an invalid input, the program should ask for an input again and again until a correct input is entered (Hint: you may use while loop here). The rules of input check are given below:

- Source String must be concatenated with the words until the word “end” (case-insensitive) is present.
- Source String must be including only letters (Both upper and lower), digits and space character. (Hint: use ASCII table)
- Search String must end with one of the ‘+’ , ‘.’ , ‘*’ characters.
- Search String cannot be empty.

Processing, Program Flow and Output

Your program should start with an introductory explanation and a prompt for the input(see sample runs). After all the inputs are entered correctly, your program should search for the string according to the rules and print out the index of the searched word.

Please note that all searches in the string are **case-sensitive**, which means that if the search string is "car", it will not match with "Car". However, for the ending words "end" and "quit", your program should perform a **case-insensitive** check. This means that if the user enters "END", "end" or “EnD” etc. to indicate the end of the string input, the program should recognize both as the end of the input. Similarly, if the user enters "QUIT", "quit" or “QuIT” etc. to indicate the end of the search queries, the program should recognize both as the end of the search process.

Note: strutils class has been added to SUCourse, so you can use it in your cpp file.

Search Rules

1. **Search String ends with a ‘+’:** The word must start with the search string.

   Example:
   ```
   Enter source string: Cars are fast END
   Enter search string: fa+
   index: 9 word: fast
   ```

2. **Search String ends with a ‘.’:** The word ends with search string.

   Example:
   ```
   Enter source string: Cars are fast END
   Enter search string: ars.
   index: 1 word: Cars
   ```

3. **Search String ends with a ‘*’:** The word that does **not** include the search string at the beginning and the end but may contain it in the middle.

   Examples:
   ```
   Enter source string: Cars are fast END
   Enter search string: as*
   index: 10 word: fast

   Enter source string: alan has a goal that study in alabama END
   Enter search string: a*
   index: 6 word: has
   index: 13 word: goal
   index: 18 word: that
   ```

4. **Search String ends with a ‘**’:** Any word containing the search string regardless of location.

   Example:
   ```
   Enter source string: Cars are fast END
   Enter search string: r**
   index: 2 word: Cars
   index: 6 word: are
   ```

An example for same search string with different search operators:

```
Enter source string: Cars are fast END
Enter search string: s+
Enter search string: s*
index: 11 word: fast
Enter search string: s.
index: 3 word: Cars
Enter search string: s**
index: 3 word: Cars
index: 11 word: fast
Enter search string: QUIT
```

An example for a different search string where search string occurs multiple times in a word:

```
Enter source string: Bu bayram da laylaylom gecti END
Enter search string: lay+
index: 13 word: laylaylom
Enter search string: lay*
Enter search string: lay**
index: 13 word: laylaylom
index: 16 word: laylaylom
Enter search string: lom+
Enter search string: lom.
index: 19 word: laylaylom
Enter search string: b+
index: 3 word: bayram
Enter search string: B+
index: 0 word: Bu
Enter search string: QUIT
```

Hint: You should use while loops and **find()** member function of the string class. Please also note that you can use **find(string searchWord, int position)** version of it to search after a given position in the string.

IMPORTANT!

If your code does not compile, you will get **zero**. Please be careful about this and double check your code before submission.

No abrupt program termination please!

You may want to stop the execution of the program at a specific place (before the end) in the program. Although there are ways of doing this in C++, it is not a good programming practice to abruptly stop the execution in the middle of the program. Therefore, your program flow should continue until the end of the main function and finish there.

Sample Runs

Below, we provide some sample runs of the program that you will develop. The *italic* and **bold** phrases are inputs taken from the user. You have to display the required information in the same order and with the same words and characters as below.

Sample Run 1 (Normal Run)

```
Enter source string: Cars are fast END
Enter search string: fast+
index: 9 word: fast
Enter search string: r*
index: 2 word: Cars
index: 6 word: are
Enter search string: s.
index: 3 word: Cars
Enter search string: r+
Enter search string: s**
index: 3 word: Cars
index: 11 word: fast
Enter search string: e.
index: 7 word: are
Enter search string: t.
index: 12 word: fast
Enter search string: fa+
index: 9 word: fast
Enter search string: QUIT
```

Sample Run 2

```
Enter source string: There are way too much homework in this class end
Enter search string: are+
index: 6 word: are
Enter search string: a.
Enter search string: a*
index: 11 word: way
index: 42 word: class
Enter search string: way
Enter search string: class
Enter search string: t+
index: 14 word: too
index: 35 word: this
Enter search string: too
Enter search string: this
Enter search string: th+
index: 35 word: this
Enter search string: this
Enter search string: o*
index: 24 word: homework
index: 28 word: homework
Enter search string: homework
Enter search string: homework
Enter search string: o.
index: 16 word: too
Enter search string: too
Enter search string: o**
index: 15 word: too
index: 16 word: too
index: 24 word: homework
index: 28 word: homework
Enter search string: quit
```

Sample Run 3 (Remember you should store the string up to word END)

```
Enter source string: there are good times and there are bad times END
Enter search string: are+
index: 6 word: are
index: 31 word: are
Enter search string: re*
Enter search string: re.
index: 3 word: there
index: 7 word: are
index: 28 word: there
index: 32 word: are
Enter search string: o*
index: 11 word: good
index: 12 word: good
Enter search string: oo*
index: 11 word: good
Enter search string: tim+
index: 15 word: times
index: 39 word: times
Enter search string: Tim+
Enter search string: Oo+
Enter search string: QUIT
```

Sample Run 4 (You should only ask for the search string when a valid source string is given)

```
Enter source string: invalid_input end
Enter source string: AnyPunctiation..,~#$£> END
Enter source string: + +2
'4 ' 3 END
Enter source string: Correct formatted input string 23 END
Enter search string: C+
index: 0 word: Correct
Enter search string: QUit
```

Sample Run 5 (You should only search when a suitable search operator is present. Also ask for search string until QUIT is given)

```
Enter source string: this is a long long homework eNd
Enter search string: invalid_input
Enter search string: long
Enter search string: wrongoperator$
Enter search string: qUiT
```

General Rules and Guidelines about Homework

The following rules and guidelines will be applicable to all homework unless otherwise noted.

How to get help?

You may ask questions to TAs (Teaching Assistants) or LAs (Learning Assistants) of CS201. Office hours of TAs/LAs are at the SUCourse.

What and Where to Submit

You can prepare (or at least test) your program using MS Visual Studio 2019 C++ (Windows users) or using XCode (macOS users).

- Your code will be automatically graded using SUCourse. Therefore, it is essential that you ensure your output matches the exact same outputs given in the example test cases provided by SUCourse.
- After writing your code, use the "Check" button located under the code editor in SUCourse to see your grade based on the example test cases used. This grade will give you an idea of how well your code is performing.
- Note that the example test cases used for checking your code are not the same as the ones used for grading your homework. Your final grade will be based on different test cases. Therefore, it is important that you carefully follow the instructions and ensure that your code is working correctly to achieve the best possible grade on your homework assignment.
- To submit your homework, click on the "Finish attempt..." button and then the "Submit all and finish" button. If you wish to submit again before the due date, you can press the "Re-attempt quiz" button.
- Submit your work **through SUCourse only**! You will receive no credits if you submit by any other means (email, paper, etc.).

Grading, Review and Objections

Be careful about the automatic grading: Your programs will be graded using an automated system. Therefore, you should follow the guidelines on the input and output order. Moreover, It is important to use the exact same text as provided in the example test case outputs from SUCourse. Otherwise, the automated grading process will fail for your homework, and you may get a zero, or in the best scenario, you will lose points.

Grading:

- There is NO late submission. You need to submit your homework before the deadline. Please be careful that SUCourse time and your computer time may have 1-2 minute differences. You need to take this time difference into consideration.
- Successful submission is one of the requirements of the homework. If, for some reason, you cannot successfully submit your homework and we cannot grade it, your grade will be 0.
- If your code does not work because of a syntax error, then we cannot grade it; and thus, your grade will be 0.
- Please submit your **own** work only. It is really easy to find "similar" programs!
- Plagiarism will not be tolerated. Please check our plagiarism policy given in the Syllabus.

Plagiarism will not be tolerated!

Grade announcements: Grades will be posted in SUCourse, and you will get an Announcement at the same time. You will find the grading policy and test cases in that announcement.

Grade objections: It is your right to object to your grade if you think there is a problem, but before making an objection please try the steps below and if you still think there is a problem, contact the TA that graded your homework from the email address provided in the comment section of your announced homework grade or attend the specified objection hour in your grade announcement.

- Check the comment section in the homework tab to see the problem with your homework.
- Check the test cases in the announcement and try them with your code.
- Compare your results with the given results in the announcement.

Good Luck!
Emine Beyza Çandır & Gülşen Demiröz & Erchan Aptoula