CS201 – Spring 2021-2022
Take-Home Exam 3
– CS201 Course String Matching –
Due April 30ᵗʰ, Saturday, 23:55 (Sharp Deadline)

Introduction

In this homework, you will write a console application that would take two string values as input and search one of them in the other.

The C++ program that you are going to implement will operate based on the selection number input by the user. According to the selection your program will either continue to get more inputs or exit the execution. The input can either be a source string or a search string and your program will search the latter string within the source string. If your program finds the search string in the source string it will output the index where the search string is found and display that index along with the search string to the console (all of the results will be printed out if multiple matches exist). There are some rules for the input strings and selection choices by the user. Details will be explained later in the document.

Your take-home exams will be automatically graded using GradeChecker, so it is very important to satisfy the exact same output given in the sample runs. You can utilize GradeChecker (http://learnt.sabanciuniv.edu/GradeChecker/) to check whether your implementation is working in the expected way. To be able to use GradeChecker, you should upload all of your files used in the take-home exam. Additionally, you should submit all of your files to SUCourse **without zipping** them. **Just a reminder, you will see a character ¶ which refers to a newline in your expected output.**

The name of your main source (cpp) file should be in the expected format: "SUCourseUsername_THEnumber.cpp" (all lowercase letters). Please check the submission procedures of the take-home exam, which are listed at the end of this document.

Description

First of all, your program will display the possible selections with example source and search strings in the beginning of the program, where there will be three choices. If selection 1 or 2 is input by the user, then you should take the source or search string as input respectively and wait for the next input. If the user enters 3 as the selection, then you should exit and end the program. If the user inputs a wrong selection number, then you should print an error message and continue the program waiting for the next input.

Secondly, when the user inputting the strings or selections you should do various input checks against all possible errors. These will be explained in the following section. Your program should only end if the user enters the selection 3 to exit.

Thirdly, your program will accept strings that consist of **ONLY** lowercase letters as source string. This rule holds for the search string too except for one character, which is **'?'**. Your search string may contain the **'?' (Question Mark)** character as a wild card, i.e. this character matches **any** character.

Finally, when your program matches the search string in the source, it should print the starting index point of the match and also the matched string.

Input Entry and Input Check

- Selection choice MUST be one of **{1, 2, 3}**. Any other digits or strings are invalid input. HINT: You may take the choice from the user as a string for the sake of simplicity.
- Source string
    o **MUST NOT** be empty
    o **MUST NOT** contain '?'
    o **MUST NOT** contain any character except lower case letters 'a'-'z'
- Search string
    o **MUST NOT** be empty,
    o **MUST NOT** be longer than the source string,
    o **MUST NOT** contain any character except lower case letters 'a'-'z' and '?'
    o **MUST NOT** be **ONLY** '?'s

If any of the inputs are not valid, your program should display an error message and expect a new input from the user.

Use of Functions and Other Rules

Unlike the second homework, we will not specify any functions here. But you are expected to use functions to avoid code duplication and improve the modularity of your program. **If your main function or any user-defined function is too long and if you do everything in main or in another user-defined function, your grade may be lowered.**

AND PLEASE DO NOT WRITE EVERYTHING IN MAIN AND THEN TRY TO SPLIT THE TASK INTO SOME FUNCTIONS JUST TO HAVE SOME FUNCTIONS OTHER THAN MAIN. THIS IS TOTALLY AGAINST THE IDEA OF FUNCTIONAL DESIGN AND NOTHING BUT A DIRTY TRICK TO GET SOME POINTS. INSTEAD PLEASE DESIGN YOUR PROGRAM BY CONSIDERING NECESSARY FUNCTIONS <u>AT THE BEGINNING</u>.

Try to use parametric functions and avoid inputs in functions. Do NOT use any global variables (variables defined outside the functions) to avoid parameter use.

As in the previous homework, no abrupt program termination is allowed in the middle of the program! The program flow should continue until the end of the main function.

IMPORTANT!

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

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 mentioned format.

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

Go to next page for Sample Runs

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. The introductory line (This
program bla bla bla) is intentionally left incomplete. You are expected to use your
imagination and creativity there to introduce your program. You may also change
the prompts and phrases in output lines. However, as mentioned above, the input
and the output order should remain intact.

Sample Run 1

```
This program searches a search string in a source string with the
option of wildcards.

Example inputs with their meaning in parentheses:
1 thislectureisawesome (to enter a new source string)
2 lecture (to enter a new search string)
2 t?r? (to enter a search string with wildcards)
3 (to exit the program)

Enter your choice and string: 1 changingthesourcestring
Source word has been changed to "changingthesourcestring"

Enter your choice and string: 2 the
For the source word "changingthesourcestring" and search word
"the",
"the" has been found at index 8

Enter your choice and string: 2 ing
For the source word "changingthesourcestring" and search word
"ing",
"ing" has been found at index 5
"ing" has been found at index 20

Enter your choice and string: 2 is
For the source word "changingthesourcestring" and search word
"is", no match has been found.

Enter your choice and string: 2 ?es
For the source word "changingthesourcestring" and search word
"?es",
"hes" has been found at index 9
"ces" has been found at index 15

Enter your choice and string: 2 ?e?
For the source word "changingthesourcestring" and search word
"?e?",
"hes" has been found at index 9
"ces" has been found at index 15

Enter your choice and string: 3
See you!
```

Sample Run 2

This program searches a search string in a source string with
the option of wildcards.

Example inputs with their meaning in parentheses:
1 thislectureisawesome (to enter a new source string)
2 lecture (to enter a new search string)
2 t?r? (to enter a search string with wildcards)
3 (to exit the program)

Enter your choice and string: *1 newsourcestring*
Source word has been changed to "newsourcestring"

Enter your choice and string: *1 newsourcestringv2*
Wrong input format! Try again.

Enter your choice and string: *1 newsourcestringvtwo*
Source word has been changed to "newsourcestringvtwo"

Enter your choice and string: *2 ?*
Wrong input format! Try again.

Enter your choice and string: *2 ???*
Wrong input format! Try again.

Enter your choice and string: *2 erf2*
Wrong input format! Try again.

Enter your choice and string: *2 sou*ce*
Wrong input format! Try again.

Enter your choice and string: *2 ?o*
For the source word "newsourcestringvtwo" and search word
"?o",
"so" has been found at index 3
"wo" has been found at index 17

Enter your choice and string: *3*
See you!

Sample Run 3

This program searches a search string in a source string with the option of wildcards.

Example inputs with their meaning in parentheses:
1 thislectureisawesome (to enter a new source string)
2 lecture (to enter a new search string)
2 t?r? (to enter a search string with wildcards)
3 (to exit the program)

Enter your choice and string: *2 sabanci*
Wrong input format! Try again.

Enter your choice and string: *1 helloworld?*
Wrong input format! Try again.

Enter your choice and string: *1 thisissourcestring*
Source word has been changed to "thisissourcestring"

Enter your choice and string: *2 this*
For the source word "thisissourcestring" and search word "this",
"this" has been found at index 0

Enter your choice and string: *2 ?i?*
For the source word "thisissourcestring" and search word "?i?",
"his" has been found at index 1
"sis" has been found at index 3
"rin" has been found at index 14

Enter your choice and string: *2 t?*
For the source word "thisissourcestring" and search word "t?",
"th" has been found at index 0
"tr" has been found at index 13

Enter your choice and string: *2 thisissourcestring*
For the source word "thisissourcestring" and search word "thisissourcestring",
"thisissourcestring" has been found at index 0

Enter your choice and string: *2 thisissourcestrin?*
For the source word "thisissourcestring" and search word "thisissourcestrin?",
"thisissourcestring" has been found at index 0

Enter your choice and string: *2 ?hisissourcestrin?*

```
For  the  source  word  "thisissourcestring"  and  search  word
"?hisissourcestrin?",
"thisissourcestring" has been found at index 0

Enter your choice and string: 5
Choice can be 1, 2 or 3! Try again.

Enter your choice and string: Cs201
Choice can be 1, 2 or 3! Try again.

Enter your choice and string: 3
See you!
```

Go to next page for Sample Run 4

Sample Run 4

This program searches a search string in a source string with the option of wildcards.

Example inputs with their meaning in parentheses:
1 thislectureisawesome (to enter a new source string)
2 lecture (to enter a new search string)
2 t?r? (to enter a search string with wildcards)
3 (to exit the program)

Enter your choice and string: *1 sabanciuniversity*
Source word has been changed to "sabanciuniversity"

Enter your choice and string: *2 university*
For the source word "sabanciuniversity" and search word "university",
"university" has been found at index 7

Enter your choice and string: *2 s?*
For the source word "sabanciuniversity" and search word "s?",
"sa" has been found at index 0
"si" has been found at index 13

Enter your choice and string: *2 s???*
For the source word "sabanciuniversity" and search word "s???",
"saba" has been found at index 0
"sity" has been found at index 13

Enter your choice and string: *1 university*
Source word has been changed to "university"

Enter your choice and string: *2 sabanciuniversity*
Wrong input format! Try again.

Enter your choice and string: *2 university*
For the source word "university" and search word "university",
"university" has been found at index 0

Enter your choice and string: *3*
See you!

General Rules and Guidelines about Homeworks

The following rules and guidelines will be applicable to all take-home exams, unless otherwise noted.

- How to get help?

You can use GradeChecker (http://learnt.sabanciuniv.edu/GradeChecker/) to check your expected grade. Just a reminder, you will see a character ¶ which refers to a newline in your expected output.

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

- What and Where to Submit

You should prepare (or at least test) your program using MS Visual Studio 2012 C++ (Windows users) or using xcode (macOS users).

It'd be a good idea to write your name and last name in the program (as a comment line of course). <u>Do not use any Turkish characters anywhere in your code (not even in comment parts).</u> If your name and last name is "İnanç Arın", and if you want to write it as comment; then you must type it as follows:
> *// Inanc Arin*

Submission guidelines are below. Since the grading process will be automatic, students are expected to strictly follow these guidelines. If you do not follow these guidelines, your grade will be 0.

- Name your submission file as follows:

    - <u>Use only English alphabet letters, digits or underscore in the file names</u>. <u>Do not use blank, Turkish characters or any other special symbols or characters</u>.

    - Name your cpp file that contains your program as follows:
      "**SUCourseUsername_THEnumber.cpp**"

    - Your SUCourse user name is actually your SUNet username, which is used for checking sabanciuniv emails. Do <u>NOT</u> use any spaces, non-ASCII and Turkish characters in the file name (**use only lowercase letters**). For example, if your SUCourse username is "**altop**", then the file name should be: **altop_the3.*cpp*** (please only use lowercase letters).

    - Do <u>not</u> add any other character or phrase to the file name.

- Please make sure that this file is the latest version of your take-home exam program.

- Submit your work **through SUCourse only**! You can use GradeChecker <u>only</u> to see if your program can produce the correct outputs both in the correct order and in the correct format. It will <u>not</u> be considered as the official submission. You <u>must</u> submit your work to SUCourse. You will receive no credits if you submit by any other means (email, paper, etc.).

- If you would like to resubmit your work, you should first remove the existing file(s). This step is very important. If you do not delete the old file(s), we will receive both files and the old one may be graded.

- Grading, Review and Objections

<u>Be careful about the automatic grading</u>: Your programs will be graded using an automated system. Therefore, you should follow the guidelines on the input and output order. Moreover, you should also use the same text as given in the "Sample Runs" section. Otherwise, the automated grading process will fail for your take-home exam, and you may get a zero, or in the best scenario, you will lose points.

<u>Grading</u>:

- There is NO late submission. You need to submit your take-home exam before the deadline. Please be careful that SUCourse time and your computer time <u>may</u> have 1-2 minutes differences. You need to take this time difference into consideration.

- Successful submission is one of the requirements of the take-home exam. If, for some reason, you cannot successfully submit your take-home exam 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 <u>only</u>. It is really easy to find "similar" programs!

- Plagiarism will not be tolerated. Please check our plagiarism policy given in the [Syllabus](#) or on the [course website](#).

Plagiarism will not be tolerated!

<u>Grade announcements</u>: 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.

<u>Grade objections</u>: 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 take-home exam from the email address provided in the comment section of your announced take-home exam grade or attend the specified objection hour in your grade announcement.
- Check the comment section in the take-home exam tab to see the problem with your take-home exam.
- Download the file you submitted to SUCourse and try to compile it.
- 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!
E. Beyza Çandır & CS201 Instructors