CS201 – Spring 2015-2016 - Sabancı University

Homework #4: Steganalysis

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

Motivation

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 required.

Description

In this homework, you will write a simple program that extracts information from a given scrambled string. The art of retrieving of hidden information from texts or images is called *steganalysis*. The program removes certain characters from a scrambled string in order to generate an extracted text. Then, the program will delete user-given characters from the extracted text to retrieve the hidden information. To perform these operations, string member functions, loops and non-void functions will be used in the program. Of course, you may need to use the topics that we have seen before such as if, void functions, etc.

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.

IMPORTANT!

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

Inputs

There are two inputs respectively:

- Scrambled Text (string)
- Characters to Delete (string)

There should be no spaces in the string input (Because if there is/are space(s) in the input, it will be considered as multiple strings, and therefore you cannot input it into a single string variable). While writing your program, you may assume that user will not enter spaces. Also note that, the order of inputs should be the same as specified here; *Scrambled Text, Characters to Delete.*

Input Check

Only *Scrambled Text* must be checked. In the case of taking invalid input, the program should ask for that input again until a correct input is entered (Hint: while loop). The rule for input check is given below:

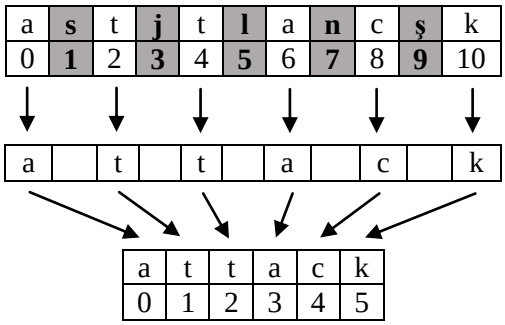
- Scrambled Text
    - Length of String: The length of string shouldn't be smaller than 40 characters.

Processing, Program Flow and Output

Your program should start with an introductory explanation and a prompt for the input. After all the inputs are entered correctly, your program should first extract certain characters using the given pattern below. Then, your program should use the *Extracted Text* string and delete every given character in the *Characters to Delete* string.

Extraction Pattern: Your program should take every even indexed (0, 2, 4, 6, 8, 10, 12, …) character in the *Scrambled Text* input.

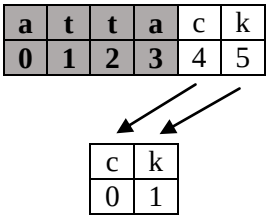
- Scrambled Text: "astjtlancşk"
  Extracted Text: "attack"



- You should output this *Extracted Text* to console.

- After this step you should delete any characters found in the input string *Characters to Delete*. Note that although you get a string from user you need to treat each character in the input string individually.

- Extracted Text: "attack"
  Characters to Delete: "at"



- You should not display result of each delete operation but only the final output. See sample runs for more examples.

Searching the string to delete characters, input check and generation of substrings require using some of the string member functions (like `length`, `find`, `rfind`, `substr`, `at`, etc.) that we covered in class.

No abrupt program termination please!

You may want to stop the execution of the program at a specific place 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.

Important Remarks

Your program must be modular and you should avoid code duplication. Thus you have to show your ability to use functions in an appropriate way. This will affect your grade. In general, **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. (Hint: implement string extraction and character deletion in separate functions)**

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 AT THE BEGINNING.

Try to use parametric and non-void functions wherever appropriate. Do NOT use any global variables (variables defined outside the functions) to avoid parameter use.

In this homework (and in the coming ones) you are not allowed to use instructions such as “exit” and “goto”. These cause difficulty to control the flow of your programs. Thus we do not approve using them. You are also not encouraged to use “break” and “continue”. The use of “break” and “continue” prevent you from forming good readable loop conditions and hence prevent you from learning how to form good loops. Think cleverly in order not to use any of these instructions. If you don't know these commands, do not even try to learn them (we will explain “break” in class).

Please remark that there are two inputs described above in the Inputs part. While input order should be exactly the same as the sample runs, also no other input is allowed (such as a name for the introduction or anything else not mentioned in this homework specification). Since your submissions are processed automatically, extra inputs cause problems and delays in the processing and grading. If you do not follow this rule, your grade may be lowered.

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 should follow the input order in these examples and the prompts your program will display must be **exactly the same** as in the following examples.

Sample Run 1

```
This program bla bla bla...
Scrambled Text: aqtstvagcrkepqecavrelqhsaarcbvobriscuksuy
Characters to be deleted: yus

Extracted String: attackpearlharborsusy
Hidden Message: attackpearlharbor
```

Sample Run 2

```
This program bla bla bla...
Scrambled Text: aodshoaih
Input string must be longer than 40 characters
Scrambled Text: dsfklngnniewo
Input string must be longer than 40 characters
Scrambled Text: kjkfnbknkbfkgnbkf
Input string must be longer than 40 characters
Scrambled Text: tuaprlgseftbancmqkulisrdehdkmşeieytwaetc116109098
Characters to be deleted: 8

Extracted String: targetacquiredmeetat16008
Hidden Message: targetacquiredmeetat1600
```

Sample Run 3

```
This program bla bla bla...
Scrambled Text: puapylms80efnbtnwgjmikl1lsbsjdehdkoojşnie
Characters to be deleted: 8vj

Extracted String: paym8entwjillbjedojne
Hidden Message: paymentwillbedone
```

Sample Run 4

```
This program bla bla bla...
Scrambled Text: 0ac1u2b3o4nxerdda6ijrbisedsugxoqtgynttyy
Characters to be deleted: cutey0

Extracted String: 0cubonedairiesgotyty
Hidden Message: bondairisgo
```

Sample Run 5

```
This program bla bla bla...
Scrambled Text: hcysaalkfe-ilsiafsulei3eciolnlfusuimrimnesyadt!i
Characters to be deleted: yus

Extracted String: hyalf-lifue3confsirmeyd!
Hidden Message: half-life3confirmed!
```

Sample Run 6

```
This program bla bla bla...
Scrambled Text: aalrl2y7omu4rybyassee2a
Input string must be longer than 40 characters
Scrambled Text: aalrl2y7omu4rybyassee2a.r0e0b0e0lcobngg3t;o,ufsz!?
Characters to be deleted: ckm0p

Extracted String: allyourbasearebelongtous!
Hidden Message: allyourbasearebelongtous!
```

General Rules and Guidelines about Homeworks

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

How to get help?

You may ask questions to TAs (Teaching Assistants) of CS201. Office hours of TAs are at the class website. Recitations will partially be dedicated to clarify the issues related to homework, so it is to your benefit to attend recitations.

What and Where to Submit

Please see the detailed instructions below/in the next page. The submission steps will get natural/easy for later homeworks.

Grading and Objections

Careful about the semi-automatic grading: Your programs will be graded using a semi-automated system. Therefore you should follow the guidelines about input and output order; moreover you should also use same prompts as given in the Sample Runs. Otherwise semi-automated grading process will fail for your homework, and you may get a zero, or in the best scenario you will lose points.

Grading:

- Late penalty is 10% off of the full grade and only one late day is allowed.
- **Having a correct program is necessary, but not sufficient to get the full grade. Comments, indentation, meaningful and understandable identifier names, informative introduction and prompts, and especially proper use of required functions, unnecessarily long program (which is bad) and unnecessary code duplications (which is also bad) will also affect your grade.**
- Please submit your own work only (even if it is not working). It is really easy to find out "similar" programs!
- For detailed rules and course policy on plagiarism, please check out http://myweb.sabanciuniv.edu/gulsend/su_current_courses/cs-201-spring-2008/plagiarism/ and keep in mind that

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.
- Download the .zip 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.

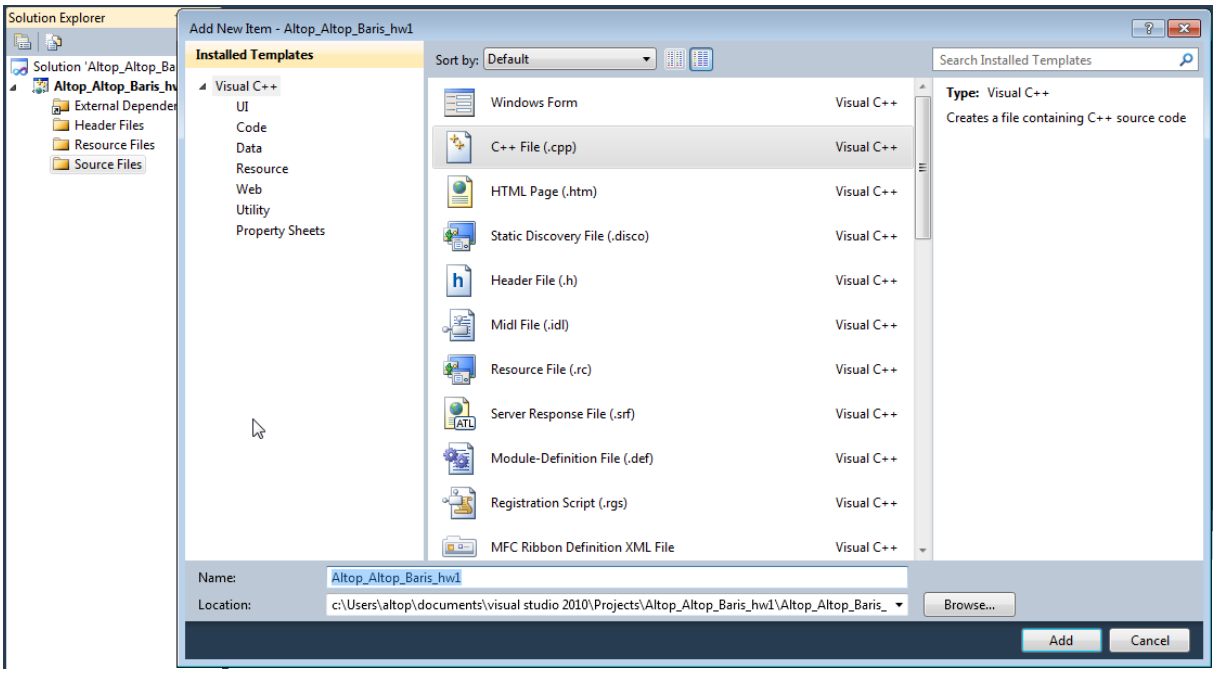
What and where to submit (IMPORTANT)

Submissions guidelines are below. Most parts of the grading process are automatic. Students are expected to strictly follow these guidelines in order to have a smooth grading process. If you do not follow these guidelines, depending on the severity of the problem created during the grading process, 5 or more penalty points are to be deducted from the grade.

Add your name to the program: It is a good practice to write your name and last name somewhere in the beginning program (as a comment line of course).

Name your submission file:

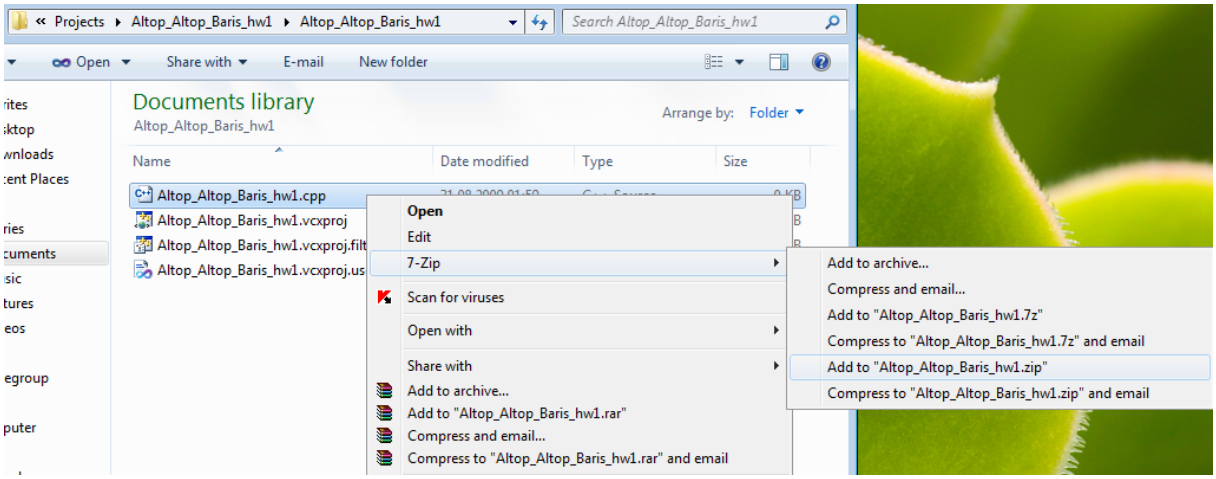
- ❑ Use only English alphabet letters, digits or underscore in the file names. Do not use blank, Turkish characters or any other special symbols or characters.
- ❑ Name your cpp file that contains your program as follows.
  "**SUCourseUserName_YourLastname_YourName_HWnumber.cpp**"



- Your SUCourse user name is actually your SUNet user name which is used for checking sabanciuniv e-mails. Do NOT use any spaces, non-ASCII and Turkish characters in the file name. For example, if your SUCourse user name is cago, name is Çağlayan, and last name is Özbugsızkodyazaroğlu, then the file name must be:

Cago_Ozbugsizkodyazaroglu_Caglayan_hw2.cpp

- Do not add any other character or phrase to the file name.
- Make sure that this file is the latest version of your homework program.
- Compress this cpp file using WINZIP or WINRAR programs. Please use "**zip**" compression. "rar" or another compression mechanism is NOT allowed. Our homework processing system works only with zip files. Therefore, make sure that the resulting compressed file has a zip extension.



- Check that your compressed file opens up correctly and it contains your **cpp** file. You will receive no credits if your compressed zip file does not expand or it does not contain the correct file.
- The naming convention of the zip file is the same as the cpp file (except the extension of the file of course). The name of the zip file should be as follows.

"**SUCourseUserName_YourLastname_YourName_HWnumber.zip**"

For example zubzipler_Zipleroglu_Zubeyir_hw2.zip is a valid name, but hw2_hoz_HasanOz.zip, HasanOzHoz.zip are NOT valid names.

<u>Submission:</u>

- Submit via SUCourse ONLY! You will receive no credits if you submit by other means (e-mail, paper, etc.).
  1) Click on "Assignments" at CS201 SUCourse (not the CS201 web site).
  2) Click Homework 2 in the assignments list.
  3) Click on "Add Attachments" button.
  4) Click on "Browse" button and select the zip file that you generated.
  5) Now, you have to see your zip file in the "Items to attach" list.
  6) Click on "Continue" button.
  7) Click on "Submit" button. We cannot see your homework if you do not perform this step even if you upload your file.

Resubmission:

- After submission, you will be able to take your homework back and resubmit. In order to resubmit, follow the following steps.
    1) Click on "Assignments" at CS201 SUCourse.
    2) Click Homework 2 in the assignments list.
    3) Click on "Re-submit" button.
    4) Click on "Add/remove Attachments" button
    5) Remove the existing zip file by clicking on "remove" link. This step is very important. If you do not delete the old zip file, we receive both files and the old one may be graded.
    6) Click on "Browse" button and select the new zip file that you want to resubmit.
    7) Now, you have to see your new zip file in the "Items to attach" list.
    8) Click on "Continue" button.
    9) Click on "Submit" button. We cannot see your homework if you do not perform this step even if you upload your file.

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.

Good Luck!
Ece Egemen and Gülşen Demiröz