EECE 230 Introduction to Programming, Sections 3,4, and 12
Programming Assignment 5

Tue Oct 30, 2012

- This programming assignment consists of 3 problems + 1 optional.
- It is due on Tue Nov 6 in the Lab at 5 pm.
- Related reading: I/O and Files, Strings.
- *Lab structure and regulations:*
  - ⋆ The 3 hours Lab session is on Tuesdays in Lab rooms 1,2, and 5 from 2:00 pm to 5:00 pm. It consists of three parts:
    - *Occasional Solving Session (not graded but attendance mandatory)*
    - *Programming Assignment (graded)*
      Programming Assignments will be posted on Moodle on weekly basis. Typically, a Programming Assignment requires much more than the time allocated for this part in the Lab, so you are supposed to complete the major part of the assignment at home. The Lab instructor will grade your assignment and can help you with the problems you are facing.
    - *Occasional graded weekly quiz*
  - ⋆ You are supposed to submit your own work. Cheating will not be tolerated and will be dealt with severely: zero grades on the programming assignments, disciplinary committee, Dean's warning.
  - ⋆ Lab attendance is mandatory. Violating this rule can lead to a failing grade.

Problem 1. File statistics.

Write a program which asks the user to enter the the name of a text file. The program is supposed to print:

1. the number of characters in the file
2. the number of words in the file
3. the number of lines in the file.

You program should store the name of the input file as a C-string, i.e., a $'\backslash 0'$-terminated char array.
Your program should print zeros if the input file does not exist.
Do not forget to close the file at the end.

(*Hints:* To count the number of lines, it is enough to count the number of $'\backslash n'$ characters. To count the number of words, it is not enough to count the number of $' '$ characters as the file may contain consecutive blank characters. One way to compute the number of words is to loop till the end of the file while reading words into a temporary C-string (using a suitable function that ignores white spaces) and incrementing a counter).

Problem 2. Deleting a character from a string.

Write a program which asks the user to enter a C-string str and an integer i. It is supposed to modify the char-array str by deleting its i'th character, where $i = 0$ corresponds to the first character, $i = 1$ corresponds to the second character, and so on ....

At the end your program is supposed to print the modified string.

For example, on input

```
Please enter a string:eece230
Please enter i:2
```

your program is supposed to store "eece230" in str, modify the content of str to "eee230", and finally print

```
eee230
```

Note that it is not enough to set the deleted character to a blank character; the length of modified string is smaller than that of the original string by one.

Problem 3. Substrings.

Write a computer program which given two C-strings as input, checks if the first is a substring of the second.

You are supposed to allow whitespaces in the input strings. You program should interpret a new line as the end of first input string and similarly for the second input string.

Note that if $s_1[0 \dots l_1 - 1]$ and $s_2[0 \dots l_2 - 1]$ are character strings of lengths l_1 and l_2 respectively, we say that s_1 is a substring of s_2 if there exists an integer i, $0 \leq i \leq l_2 - (l_1 - 1)$, such that $s_1[j] = s_2[i + j]$ for $j = 0, \dots, l_1 - 1$

For example, "obl" is a substring of "Problem 2", but "oblm" is not a substring of "Problem 2".

Your program is supposed to give a YES/NO answer only.

Hint: You need two nested loops and boolean variable.

You are supposed to use the *break* statement to break the outer loop when the first occurrence of the $s1$ in $s2$ is found (if any).

Note: To read two C-strings using cin.get, use an auxiliary variable to absorb the newline after reading the first string (as otherwise the second string will be set to the empty string due the the newline remaining in the input stream .... something annoying in C++):

```
char str1[100];
char str2[50];

cout<<"Enter first string:"<<endl;
cin.get(str1,100);

char discard;
cin.get(discard);

cout<<endl<<"Enter second string:"<<endl;
cin.get(str2,50);
```

Problem 4. (Optional) Files: find and replace words.

Write a computer program which asks the user the enter 4 C-strings: $fileName1$, $fileName2$, $word1$, and $word2$. Your program should open the file whose name is $fileName1$ as an input file (assuming it

exits), find all the occurrences of the string $word1$ in the this file, and create a new file named $fileName2$ with all the occurrences of $word1$ replaced with $word2$.

Example: assume that $fileName1$ = "*input.txt*", $fileName2$ = "*output.txt*", $word1$ = "*ile*", $word2$ = "*IeL*", and assume that the file "input.txt" consists of the text:

```
Write a computer program which asks the user the enter 4
C-strings: fileName1, fileName2, word1, and  word2.
Your program  should open the  file whose name is  fileName1
as an input file (assuming it exits), find all the occurrences
of the string word1 in the this file,  and create a  new file
named fileName2 with all the occurrences of word1 replaced
with  word2.
```

Then your program is supposed to create a file called "output.txt" and store in it

```
Write a computer program which asks the user the enter 4
C-strings: fIeLName1, fIeLName2, word1, and  word2.
Your program  should open the  fIeL whose name is  fIeLName1
as an input fIeL (assuming it exits), find all the occurrences
of the string word1 in the this fIeL,  and create a  new fIeL
named fIeLName2 with all the occurrences of word1 replaced
with  word2.
```

Hint: Read all the input file into a character array (with enough memory allocation, say 10000 bytes) and use/modify your code from the substrings problem (Problem 3 above).