School of Electrical and Information Engineering
University of the Witwatersrand, Johannesburg
ELEN2020A - Software Development I

Lab 4 - Strings

Lab 4 consists of 2 tasks. Each week one task must be completed and submitted. Check the due date of each task on Ulwazi. Note that the following test and submission instructions apply for all tasks. Note that the following test and submission instructions apply for all tasks.

Testing your code

You should refer to the provided set of inputs and outputs listed in the lab brief to determine if your code is running correctly. Given the set of inputs your code should be able to reproduce the output provided. Make sure to test it with additional values not in the lab brief to ensure it is working correctly.

Submission instructions

Your source code (.cpp) must be submitted to the *Lab 4 - task X* link under *Assignments* on Ulwazi (where **X** is the task number). **Only submit a zip of your source code and flow chart diagram.**

It is important that you follow **good programming practices** in your code. This includes

- indent code correctly (use the AStyle formatting tool in Code::Blocks),
- add necessary comments and choose meaningful names of variables in your code,
- add the date and author at the top of your source code. Add the following comment at the start of each source code file.

```
1 // Student number:
2 // Date:
```

Note: the following information may be useful.

Table 1: **`#include <cctype>`**

Type	Method	Description
int	**isalnum(c)**	true if **`c`** is a letter or digit.
int	**isalpha(c)**	true if **`c`** is a letter.
int	**isblank(c)**	true if **`c`** is a blank or tab.
int	**isdigit(c)**	true if **`c`** is a digit.
int	**islower(c)**	true if **`c`** is a lowercase letter.
int	**isupper(c)**	true if **`c`** is a uppercase letter.
int	**isspace(c)**	true if **`c`** is a whitespace character (space, tab, vertical tab, carriage return, newline).
int	**tolower(c)**	returns lowercase version of **`c`** if there is one, otherwise it returns the character unchanged.
int	**toupper(c)**	returns uppercase version of **`c`** if there is one, otherwise it returns the character unchanged.

Task 1 (week 1) - Reverse words

Encryption is the process of converting data to an unrecognisable form. It is commonly used to protect sensitive information so that only authorized parties can view it. One such encryption method is to write each word with its letters in the reverse order.

For example given the plaintext message

```
This problem is too easy for me. I am an amazing programmer. Do you agree?
```

To encrypt this plaintext we take each word and reverse it to find the encrypted text. The first word 'This' will become 'sihT', the second word 'problem' will become 'melborp'. Repeating this process will result in the encrypted message being

```
sihT melborp si oot ysae rof em. I ma na gnizama remmargorp. oD uoy eerga?
```

You are required to write a program that

- reads in from a file named **`input.txt`** the plaintext (on an unknown number of lines),
- encrypts the plaintext using a function written by you,
- leaves all other (non-alphabetic) characters in the plaintext unchanged,
- and writes the encrypted message to a file named **`output.txt`**.

Note that any text that appears on a new line in the input should also appear on a newline in the output (see the examples below).

Example 1

An example of the input and output file can be seen in Listings 1 and 2.

Listing 1: **input.txt**

```
This problem is too easy for me. I am an amazing programmer. Do you agree?
```

Listing 2: **output.txt**

```
sihT melborp si oot ysae rof em. I ma na gnizama remmargorp. oD uoy eerga?
```

Example 2

Another example of the input and output file can be seen in Listings 3 and 4.

Listing 3: **input.txt**

```
Expecting the world to treat you
fairly because you are a good person
is a little like expecting the bull not to
attack you because you are a vegetarian.
```

Listing 4: **output.txt**

```
gnitcepxE eht dlrow ot taert uoy
ylriaf esuaceb uoy era a doog nosrep
si a elttil ekil gnitcepxe eht llub ton ot
kcatta uoy esuaceb uoy era a nairategev.
```

Task 2 (week 2) - Pig Latin

Write a program that converts an input string into the pig Latin form. The rules for converting a string into pig Latin form are as follows:

1. If the string begins with a vowel, add the string "-way" at the end of the string. For example, the pig Latin form of the string "eye" is "eye-way".
2. If the string does not begin with a vowel, first add "-" at the end of the string. Then rotate the string one character at a time; that is, move the first character of the string to the end of the string until the first character of the string becomes a vowel. Then add the string "ay" at the end. For example, the pig Latin form of the string "There" is "ere-Thay".
3. Strings such as "1234" contain no vowels. The pig Latin form of the string "1234" is "1234-way". That is, the pig Latin form of a string that has no vowels in it is the string followed by the string "-way".

Your program must

- read in text from a file named **`input.txt`**,
- convert the input text to pig Latin using a function written by you and
- write the converted text to an output file named **`output.txt`**.

Example 1

An example of the input and output file can be seen in Listings 5 and 6.

Listing 5: **`input.txt`**

```
Thatll do Pig Thatll do
Farmer Hogget
```

Listing 6: **`output.txt`**

```
atll-Thay o-day ig-Pay atll-Thay o-day
armer-Fay ogget-Hay
```

Example 2

Another example of the input and output file can be seen in Listings 7 and 8.

Listing 7: **`input.txt`**

```
You got a fast car
I want a ticket to anywhere
Maybe we make a deal
Maybe together we can get somewhere
Anyplace is better
Starting from zero got nothing to lose
Maybe well make something
Me myself I got nothing to prove
Tracy Chapman
```

Listing 8: **output.txt**

```
ou-Yay ot-gay a-way ast-fay ar-cay
I-way ant-way a-way icket-tay o-tay anywhere-way
aybe-May e-way ake-may a-way eal-day
aybe-May ogether-tay e-way an-cay et-gay omewhere-say
Anyplace-way is-way etter-bay
arting-Stay om-fray ero-zay ot-gay othing-nay o-tay ose-lay
aybe-May ell-way ake-may omething-say
e-May elf-mysay I-way ot-gay othing-nay o-tay ove-pray
acy-Tray apman-Chay
```