

## Homework 1: Editor with Doubly Linked Lists

Deadline: Sept. 23, 2014

### 1. INTRODUCTION

You will write a C++ program that will implement basic functionality of an editor. Since creating an interactive editor would be too time consuming, you will program a command-line editor which does only insertion and deletion of words. This editor is similar to early editors and UNIX utilities.

You must use **doubly linked lists** to store a list of words; you cannot use arrays. Words will be inserted and deleted with dynamic memory manipulation via linked lists. The first major requirement is that your program produces correct results. The second requirement is that it manages memory efficiently. You must develop your own C++ classes to solve the homework. Therefore, you are **not allowed to use the list implementation in the STL Library**.

### 2. INPUT & OUTPUT

The input of the program is a text file with multiple commands to be executed by the editor, one command per line. The commands are defined as follows:

- **Read** [*file name*]  
Read the specified file from disk and insert each word in the file into a doubly linked list. The file specified in this command will be a text file, with an unbounded number of different words, separated by spaces, commas or periods. The input file can have several lines terminated by carriage returns. You can assume words will be at most **50** characters long and no special characters will appear in front or in the middle of each word.

**Example:**

```
Read input.txt
```

*input.txt:*

```
Microsoft has many many products,  
such as Windows
```

- **Print** [*forward / backward*]  
Print all the elements in the linked list to screen, one element per line. You should start from the head element if parameter “forward” is specified, from the tail element otherwise.

**Example 1:**

```
Print forward
```

After executing the **read** command above, the output of this command will be:

```
Microsoft
has
many
many
products
such
as
Windows
```

**Example 2:**

```
Print backward
```

The output will be:

```
Windows
as
such
products
many
many
has
Microsoft
```

- **Insert** [*word1*] [*word2*]  
Find *word2* in the doubly linked list. If *word2* is found, insert *word1* after the **first occurrence** of *word2*. Otherwise, do nothing and skip this command.

**Example:**

```
Insert yeahyeahyeah many
Print forward
```

The output will be:

```
Microsoft
has
many
yeahyeahyeah
many
products
such
as
Windows
```

- **Delete** [*word*]  
Delete the **first occurrence** of the given *word* from the doubly linked list. If the *word* cannot be found in the list, do nothing and skip this command.

**Example:**

```
Delete many
Print forward
```

Given the doubly linked list shown above, the output will be:

```
Microsoft
has
yeahyeahyeah
many
products
such
as
Windows
```

- **All other commands should be ignored.**

### 3. PROGRAM AND ARGUMENT SPECIFICATION

The main program should be called “**editor**”. The program should be able to take the first argument as input file name.

The call syntax will be like:

```
editor.exe commands.txt
```

Note that the input file name will not necessarily be the same every time, so your program shouldn't have this “commands.txt” hard coded.

The elements and commands in this program are **CASE-SENSITIVE**. “Windows” and “windows” are considered as two different words!

### 4. A SAMPLE TEST CASE

*commands.txt:*

```
Read input.txt
Insert yeahyeahyeah many
Delete many
Insert Office Windows
Insert Nokia Windows
Delete nokia
Print forward
```

*input.txt:*

```
Microsoft has many many products,
such as Windows
```

*Program call:*

```
editor.exe commands.txt
```

*Output:*

```
Microsoft
has
yeahyeahyeah
many
products
such
as
Windows
Nokia
Office
```

The element “Nokia” wasn't deleted because the command is case-sensitive.

## 5. SUBMISSION REQUIREMENTS

Your submission should be well tested before submitting under Visual Studio 2010 or later versions. You can get a copy of Visual Studio from the UH website using your cougar net username and password. The URL is <http://uh.edu/infotech/php/software/list.php>.

We use the UH blackboard system to collect your homework submissions. Before you submit your homework, please make sure to **put everything in a ZIP file** named in the form of **LastName\_PeopleSoftID\_HW1.zip**.

For example: Zhang\_1234567\_HW1.zip

The instructions about how to use the blackboard system can be found on the TA's webpage for this course: <http://www2.cs.uh.edu/~yzhang/cosc2320-f2014/>

Please submit in time, all submissions submitted after the due date will accrue a penalty of **33%** per day.

## 6. GRADING

The maximum grade for this homework is 100.

You will get 15 pts for submitting the homework in time, 10 pts if your program can be successfully compiled.

We will test your program with 5 easy test cases and 5 hard test cases. Each easy test case will worth 10 pts, and each of the hard ones will worth 5 pts.

When testing, we will compare your program's output with the standard output. Therefore **do not print any content on the screen unless required**, avoid any prompt information like "Please enter the input file name:", "The elements in the doubly linked list are:" etc.

Last but not least, **no cheating or plagiarism will be tolerated in any graded submissions.**

If you have any other questions, please send email to [zhangyiqun9164@gmail.com](mailto:zhangyiqun9164@gmail.com).