

Lab 1 – Command Line Interpreter

Due 9/1/09

Points: 20

Purpose

The purpose of the lab is to implement a basic Command Line Interpreter (CLI) /simple tokenizer in C.

Lab Description

The program will

1. Displays a welcome message
2. Repeat the following sequence until the “quit” command is entered
 1. Display “Command: “
 2. Gets a line of input
 3. Tokenize the input line
 4. If the first token of the line is “quit”
 - Then exit the loop and terminate the program
 - Else display all of the tokens each on a separate line

The following is one possible output:

```
Zeta - Version 1.0
```

```
Command: This is a simple line
```

```
This
```

```
is
```

```
a
```

```
simple
```

```
line
```

```
Command: And another
```

```
And
```

```
another
```

```
Command: quit ignore the rest of this line
```

C Implementation Issues

Use the *cin.getline* function to get a line from the keyboard. The first argument is a pointer to an input buffer where the characters will be stored. The second argument is the maximum number of characters that can be stored in the buffer.

Use *strcmp* to compare two strings.

Create a *getToken* function that will return one token at a time from the input buffer. Its signature will probably look like this:

```
char* getToken();
```

Implement an *isWhiteSpace* function that returns true if the character passed to it is white space (a blank, a tab, a line feed) and false if it is a CRLF or EOF.

The *getToken* function will move through the input buffer character by character. Each invocation of the function will ignore white spaces. When it finds the first non-white space it will copy the non-white space character to a token buffer. When a white space character is found it will put a *NULL* terminator at the end of the token buffer. The token buffer will then be returned to the calling function.

If the end of the input buffer is reached, the *getToken* function will return a *NULL* string.

Verify that your code works for the following test cases:

```
Command:  leading white space  
Command: trailing white space  
Command: leading and trailing white space
```