

Ajay Srivastava (as1877)
Srihari Chekuri (svc31)

CS 214 Assignment 1: Tokenizer Readme

The tokenizer assignment involved creating, modifying and unreferencing pointers in C. This was ultimately the biggest hurdle as we are new to the C programming language. This program takes in 2 arguments(the file name and a phrase to be tokenized). The string library in C came in handy when manipulating the strings, especially since C does not have “strings” but rather an array of chars. We made a tokenizer struct with the following attributes:

```
char *token  
char *inputString  
char *start  
char *current
```

First we allocate memory for the struct and inputString. Then we copy the original string into *inputString. Then we pass the new tokenizer struct into tkGetNextToken and create substring tokens. We set our *start and *current pointers to the very first char in *char. Then we move *current until it reaches a separator, escape character or a white space. Then we create a substring using the *start reference pointer and *current pointer. Then set our *start= *current. This substring was assigned to *token and then printed out along with its type. Then we free the memory allocated to token, inputString and the tokenizer struct. Words start with an alphabet and have alphanumerics. Octals start with a 0 and have numbers between 0-7. Hex starts with 0x or 0X and are followed by hexadecimal digits. Integers start with a digit and can be followed by any number of digits between 0-9. All the C operators are checked for. All numbers appearing after a decimal are part of a floating point number. It will also check whether a C keyword is in the word tokens. Also it skips over all comments in the code.