Zack Colello & Anna Genke 17 September 2014
Programming Assignment 1: Tokenizer.

Included in this package is a file tokenizer.c, testcases.txt and a makefile.

tokenizer.c creates one new struct, TokenizerT, which consists of 2 char array pointers, input and delimiters. It has seven functions and a main function. The seven functions are:

1. TKCreate- takes in 2 char arrays, specifically argv[1] and argcv[2], and copies them into 2 new chuncks of allocated memory and points the TokenizerT's delims and input to the allocated memory, respectly
2. Translate- take in a string and replaces any two characters which make up an escape with their literal equivalent, by assigning the array element with the hexadecimal equivalent, casted to an int.
3. TrimBuffer- takes in a Buffer string, ending with '\0' and puts the string into a chunk of memory the correct size for the string.
4. TKGetNextToken - takes in a TokenizerT struct and tokenizes the string 'tk.inputs' according to the set of delimiters in the string 'tk.delims'
5. isDelimiter- takes in the string tk.delimiters and a character held in tk.input[i] and determines if that character is a delimiter. It returns a 1 if the character is a delimiter and a 0 if it is not.
6. TranslateHexString- takes in a string and iterates through it, adding the characters of the string it takes in, into a buffer. If it encounters any of the escape characters it adds 6 characters '[', '0', 'x', 'X', 'X', ']' where the capatial Xs stand for the 2 hexadecimal digits for the escape characters.
7. TKDestroy- takes in a tokenizer and frees all malloced memory for tokenizer, tokenizer.delimiters and tokenizer.input

testcases.txt contains the test cases used to test the project.

makefile includes a make instruction with the tags -Wall -o -tokenizer -tokenizer.o
and a clean instruction.