Bilal Quadri & Yvgeniy Demo

Our program takes exactly two inputs as specified in the project description. We then run TKCreate with the input that creates and mallocs a struct with 3 pointers and returns the pointer to the struct. Memory is allocated for two of these and the arguments to TKCreate are strcpy'ed over. The third pointer is then set to point at where the allocated copy of the ts was set. This is the pointer we use to iterate over when

We then keep running the TKGetNextToken in a while loop until it hits the null terminator. TKGetNextToken first checks if the current token is a delimiter and it continues to move on to the next one until it's not. Afterwards, it also checks if it's pointing at NULL, in which case, it will simply return NULL. We then find the first occurrence of a delimiter using strpbrk and create a pointer to it. If the endOfToken is null that means the whole string from the current token needs to be returned. Otherwise we strncpy from the stream to the endOfToken and then more the stream forward up to the pointer. We free each returned token after printing it in main.

Finally, we run TkDestory which frees up all the memory allocated in the struct. First, it frees tk–>head and tk –>delims. In the end, it free the TokenizerT pointer itself.

As far as special characters are concerned our program works when they are input literally as special characters, as opposed to a literal backslash character followed by something.