There are several functions in this project:

1. TokenizerT *TKcrete (char * ts) { }

This function creates a new TokenizerT object for a taken given string. The main function will call this function to generate a token with the input string.

2. Void TKDestroy (Tokenizer T * tk) { }

The function destroys a TokenizerT object and frees the memory.

3. int invalid ()

The function will decide whether a character is invalid or not based on those parameters that passed on from TkGetNextToken

4. char * TkGetNextToken (TokenizerT * tk) { }

This function returns the next token from the token stream as a character string. If the input String is “123 12.34 apple”. The returned string will be “123”, “12.34” and “apple” in sequence.

This function also find out all the invalid character and send them to indentify function. For example, if the input is “1234a1234” then the output is

Decimal “1234”

Invalid “a”

Decimal “1234”

5. int malChar (char ptr){ }

This function will set the global char to the malform char of token. If there’s one escape character in a token, then the token

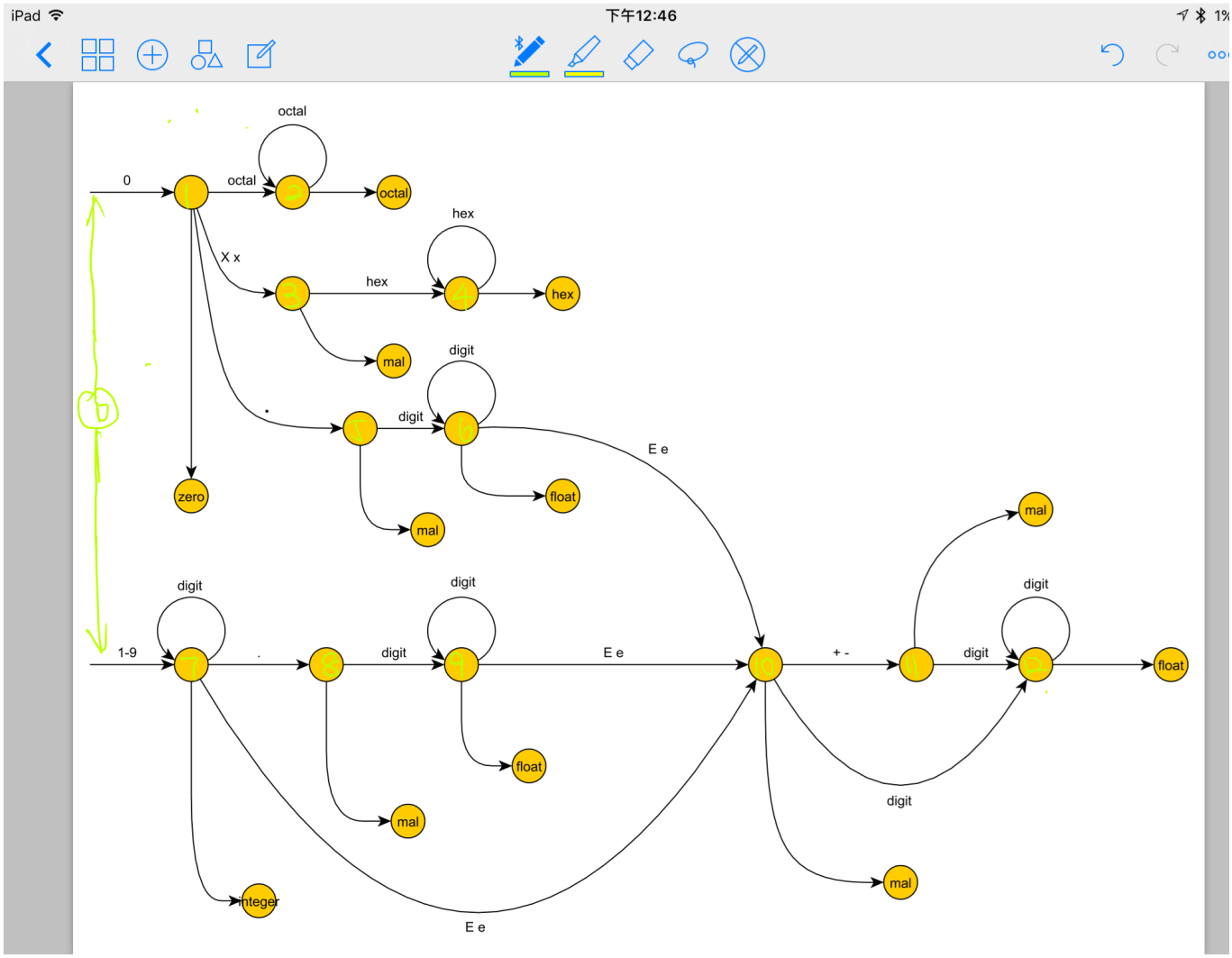
will be identified as malform.

6. int state 0 {

} .....

int state 12 { }

These 13 functions each play a part in the finite state map. They each will return an integer which indicates the next state the token will be classified. You can find more information about these 12 states in the picture below.



7. int IndentifyToken (char *ptr) { }

This function aims to identify the type of the token based on the integer returned.

8. char* TKGetState () { }

This function is for output purpose. It will get the type of the token according to the final state the token ends. For example, if the final state is “state 1”, it will return “zero”