Git: https://github.com/916-Preda-Andrei/Lftc_lab/tree/Lab3

The Scanner uses Regex for matching the strings for constants and identifiers.

String constant Regex: ^\"[a-zA-Z0-9]*\"$
This regex checks if the whole string matches with \" (which is the escaped quotes and by using \ I ensure the string isn't prematurely terminated). Afterwards the string contains as many lowercase and uppercase letters and digits as possible. This group is then captured in code to directly operate with it.

Int constant Regex: ^([1-9][0-9]*|0)$
This regex checks if the beginning of the string matches with a string, surely contains a digit from 1 to 9 and as many digits (possibly none) from 0 to 9. Either all that or a simple 0.

Identifier Regex: ^[a-zA-Z][a-zA-Z0-9]*$
This regex checks if the beginning of the string matches with a string that surely contains a character that is a lowercase/uppercase letter, then as many, possibly no characters that are either lowercase/uppercase letters, or digits. All this is again captured in a group to use further.

My PIF is built by comparing the token with the word that was read. The value of the pair is another pair that contains its position in the hash table that represents which bucket and which position inside that bucket represents your token.

I parsed the program by first reading the entire file into a list of strings, representing a line, and then simply processing word by word this line. Then, I check if this word is a token, an identifier, an int constant or a string constant. I used Regex to do the actual checking.