

```
# FILE: tokenizer.py
# FINAL PROJECT
# Professor Ahmadnia
# Group: Kevin Vuong, Anika Corpus, Christopher Grant
# Description: Provides a function to parse the source code for tokens.

import re

from token import *

def tokenizer(string):
    """
    Creates a list of tokens from the input string

    :param string: The source code
    :return: A list of tokens. Returns -1 if a token is not recognizable.
    """
    content_lines = string.split('\n')

    token_list = []
    reserved_pattern = r'^(PROGRAM|INTEGER|PRINT|BEGIN|END\.)$'
    symbol_pattern = r'^(=|\/|\*|\+|\-|\;|\:|\,|\(|\)|\$)'
    number_pattern = r'^(\+|\-)?[0-9]+$'
    identifier_pattern = r'^(P|Q|R|S)+(P|Q|R|S|[0-9])*$'

    line_number = 1 # used to keep track of line number
    for line in content_lines:
        words_list = line.split()
        for word in words_list:
            if re.match(reserved_pattern, word): # Check for reserved word token
                reserved = Token('RESERVED', word, line_number)
                token_list.append(reserved)

            elif re.match(number_pattern, word): # Check for number token
                for symbol in word:
                    if re.match('(\+|\-)', symbol):
                        sign = Token('SIGN', symbol, line_number)
                        token_list.append(sign)
                    elif re.match('[0-9]', symbol):
                        digit = Token('DIGIT', symbol, line_number)
                        token_list.append(digit)

            elif re.match(symbol_pattern, word): # Check for symbol token
                symbol = Token('SYMBOL', word, line_number)
                token_list.append(symbol)

            elif re.match(identifier_pattern, word): # Check for identifier token
                for symbol in word:
                    if re.match(r'(P|Q|R|S)', symbol):
                        _id = Token('ID', symbol, line_number)
                        token_list.append(_id)
                    elif re.match(r'[0-9]', symbol):
                        more_id_digit = Token('MORE_ID_DIGIT', symbol, line_number)
                        token_list.append(more_id_digit)
            else:
                print('Unknown word: ' + word, 'on line', line_number)
                return -1

        line_number += 1

    # For debug purposes
    # for token in token_list:
    #     print(token.get_type(), ':', token.get_value(), ':', token.get_line_num())

    return token_list
```