# COMPILER DESIGN LAB 4

import re

*def* is\_id(*tok*):

    return re.match(*r*'[\_a-zA-Z][\_a-zA-Z0-9]\*', *tok*) != None

*def* is\_num(*tok*):

    return re.match(*r*'\b\d+\b', *tok*) != None

*def* tokenizer(*s*):

    kw = {'int', 'float', 'char', 'if', 'else', 'while', 'for'}

    op = {'+', '-', '\*', '/', '=', '==', '++', '--', '%', '&&', '||', '!', '>', '<', '>=', '<='}

    dl = {';', ',', '(', ')', '{', '}'}

    id\_limit = 31

*s* = re.sub(*r*'//.\*', '', *s*)

*s* = re.sub(*r*'/\\*.\*?\\*/', '', *s*, *flags*=re.DOTALL)

*s* = re.sub(*r*'\s+', ' ', *s*).strip()

    tokens = re.findall(*r*'==|!=|<=|>=|\+\+|--|[+\-\*/%=&|!<>]=?|[;,(){}]|\b\d+\b|[\_a-zA-Z][\_a-zA-Z0-9]\*', *s*)

    for tok in tokens:

        if tok in kw:

            print("Keyword:", tok)

        elif is\_id(tok):

            if len(tok) > id\_limit:

                print("Warning!!!!")

                print("Identifier:", tok[:id\_limit], "(truncated)")

            else:

                print("Identifier:", tok)

        elif is\_num(tok):

            print("Number:", tok)

        elif tok in op:

            print("Operator:", tok)

        elif tok in dl:

            print("Delimiter:", tok)

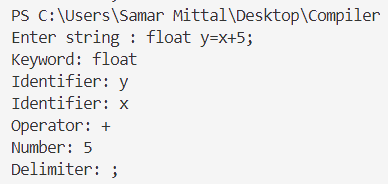
        else:

            print("Unrecognized token:", tok)

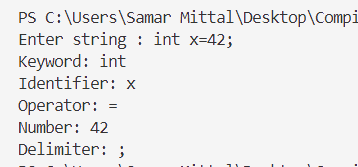
# Example usage

s=input("Enter string : ")

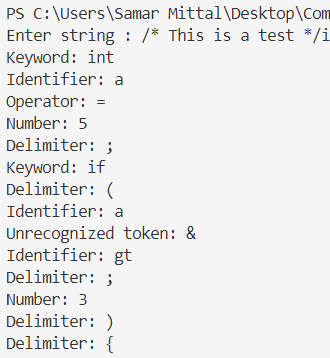
tokenizer(s)



TEST CASE 1



TEST CASE 2



TEST CASE 4

TEST CASE 3

