University of Asia Pacific

Department of Computer Science & Engineering

Compiler Design Lab
CSE 430

Submitted to:

Baivab Das
Lecturer
CSE, University of Asia Pacific

Submitted by:

Md. Farhad
20101073
Section: B1

Problem

Write a program for lexical analysis that takes input from a file or keyboard and specifies each word or character into the tokens below. The lexical analyzer should ignore redundant spaces, tabs, and newlines. It should also ignore comments and identify duplicate identifiers.

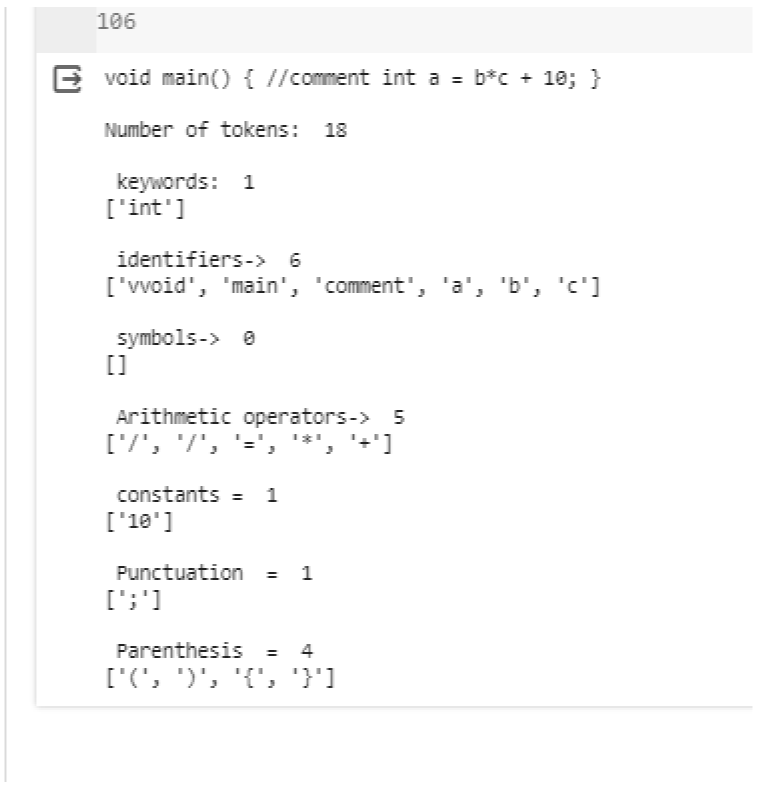
Sample Input (Console Input/ File Input):

```
void main()
{
int a, b, c;
//comment
int a = b*c + 10;
}
```

Code

```
import re

test = input()

operators = ['+', '-', '*', '/', '=']
keywords = ['auto', 'break', 'case', 'char', 'const',
'continue', 'default', 'do', 'double', 'else', 'enum', 'extern',
'float', 'for', 'goto', 'if', 'int', 'long', 'register',
'return', 'short', 'signed', 'sizeof', 'static', 'struct',
'switch', 'typedef', 'union', 'unsigned', 'void', 'volatile',
'while']
punctuation = [' ', ' ', '.', ',', ';']
parenthesis = ['(', ')', '{', '}', '[', ']']
symbols = ['!', '@', '#', '$', '%', '&', '^']

opa = []  # list define for arithmetic operators
key = []
pun = []
par = []
sym = []
ide = []
con = []
tokens = []
Str = False  # check string
Word = False  # check word
Cmt = 0  # add number of comment
token = ''  # space of token

# Check given input have any string, word, comment, space,
symbol
for i in test:
    if i == '/':
        Cmt = Cmt + 1
    elif Cmt == 2:
        if i == '\n':
            token = ''
            Cmt = 0

# checking given input have any string if have then add string
in token[] list
    elif i == '"' or i == "'":
        if Str:
            tokens.append(token)
            token = ''
        Str = not Str
    elif Str:
        token = token + i

# checking given input have any symbol if have then add symbol
in token[] list
    elif i in symbols:
        tokens.append(i)

# checking given input have any number or word if have then add
this number and word in token[] list
    elif i.isalnum() and not Word:
# isalnum() is a built-in Python function that checks whether
all characters in a string are alphanumeric.
        Word = True
        token = i

# checking given input have any punctuation and parenthesis or
at a time operator if have then added in token[] list
    if (i in punctuation) or (i in operators):
        if token:
            tokens.append(token)
            token = ''
        if not (i == ' ' or i == '\n' or i == '  '):
            tokens.append(i)
    elif (i in parenthesis) or (i in operators):
        if token:
            tokens.append(token)
            token = ''
        if not (i == ' ' or i == '\n' or i == ' '):
            tokens.append(i)
    elif Word:
        token = token + i

# token to another list such as symbol, keyword etc
for token in tokens:
    if token in symbols:
        sym.append(token)
    elif token in operators:
        opa.append(token)
    elif token in keywords:
        key.append(token)
    elif re.search("^[_a-zA-Z][_a-zA-Z0-9]*$", token):
        ide.append(token)
    elif token in punctuation:
        pun.append(token)
    elif token in parenthesis:
        par.append(token)
    else:
        con.append(token)

print("\nNumber of tokens: ", len(tokens))

print("\n keywords: ", len(key))
print(key)

print("\n identifiers-> ", len(ide))
print(ide)

print("\n symbols-> ", len(sym))
print(sym)
print("\n Arithmetic operators-> ", len(opa))
print(opa)

print("\n constants = ", len(con))
print(con)

print("\n Punctuation  = ", len(pun))
print(pun)

print("\n Parenthesis  = ", len(par))
print(par)
```

observed output:



```
106
void main() { //comment int a = b*c + 10; }

Number of tokens:  18

 keywords:  1
['int']

 identifiers->  6
['vvoid', 'main', 'comment', 'a', 'b', 'c']

 symbols->  0
[]

 Arithmetic operators->  5
['/', '/', '=', '*', '+']

 constants =  1
['10']

 Punctuation  =  1
[';']

 Parenthesis  =  4
['(', ')', '{', '}']
```