SP Assignment 6

---------------------------------------------

Name: Yash Oswal

Div: B Roll:38

SRN: 201901226

---------------------------------------------

Input Code:

#include <stdio.h>

main()

{

int var ;

var = 10 ;

printf ( " The value of a is %d ", var ) ;

return 0 ;

}

Code:-

import re

f = open('input.txt','r')

output = []

operators = { '=': 'Assignment Operator','+': 'Additon Operator', '-' : 'Substraction Operator', '/' : 'Division Operator', '\*': 'Multiplication Operator', '++' : 'increment Operator', '--' : 'Decrement Operator'}

optr\_keys = operators.keys()

header = {'.h': 'header file'}

header\_keys = header.keys()

sp\_header\_files = {'<stdio.h>':'Standard Input Output Header','<string.h>':'String Manipulation Library'}

macros = {r'#\w+' : 'macro'}

macros\_keys = macros.keys()

datatype = {'int': 'Integer','float' : 'Floating Point', 'char': 'Character','long': 'long int'}

datatype\_keys = datatype.keys()

keyword = {'return' : 'keyword that returns a value from a block','printf':'Print the string'}

keyword\_keys = keyword.keys()

delimiter = {';':'terminator symbol semicolon (;)','{' : 'Block', '}':'Block','(':'Open block',')':'Close Block'}

delimiter\_keys = delimiter.keys()

builtin\_functions = {'printf':'printf prints its argument on the console'}

non\_identifiers = ['\_','-','+','/','\*','`','~','!','@','#','$','%','^','&','\*','(',')','=','|','"',':',';','{','}','[',']','<','>','?','/']

numerals = ['0','1','2','3','4','5','6','7','8','9','10']

# Flags

dataFlag = False

i = f.read()

cc=0

ic=0

dc=0

id=[]

count = 0

program = i.split('\n')

output.append(['Line','Lexeme','Token','Token Value'])

for line in program:

count = count+1

tokens = line.split(' ')

for token in tokens:

if '\r' in token:

position = token.find('\r')

token=token[:position]

if token in optr\_keys:

output.append([count,token,operators[token],list(optr\_keys).index(token)])

if token in macros\_keys:

output.append([count,token,macros[token],list(macros\_keys).index(token)])

if '.h' in token:

output.append([count,token,'Identifier',list(sp\_header\_files.keys()).index(token)])

if '()' in token:

output.append([count,token,'Function',0])

if token in id:

output.append([count,token,'Identifier',id.index(token)])

ic+=1

if dataFlag == True and (token not in non\_identifiers) and ('()' not in token):

output.append([count,token,'Identifier',ic])

id.append(token)

ic+=1

if token in datatype\_keys:

output.append([count,token,datatype[token],list(datatype\_keys).index(token)])

dataFlag = True

if token in keyword\_keys:

output.append([count,token,'Keyword',list(keyword\_keys).index(token)])

if token in delimiter\_keys:

output.append([count,token,"Delimiter",list(delimiter\_keys).index(token)])

if '#' in token:

match = re.search(r'#\w+', token)

output.append([count,token,'Keyword',ic])

ic+=1

if token in numerals:

output.append([count,token,'Constant',cc])

cc+=1

dataFlag = False

print(output[0],"\n-------------------------------------")

for i in range(1,len(output)):

print(output[i])

f.close()

Output:

