Name = Harshit Yadav

Semester = 2

Section = CSE

Registration No = 190913017

Course Name = ASS W Lab

Signature = [illegible]

Date = June, 3, 2020

~~Assig~~ CSE Lab Test

Q.1 A) write a python program to identify relational operators from a ~~c/~~ c file.

```
import ply.lex as lex
import ply.yacc as yacc
import sys
tokens = ['FLOAT', 'INT', 'NAME', 'PLUS', 'MINUS', 'DIVIDE', 'MULTIPLY'
         ~~EQ~~ 'EQUAL']

~~t_PLUS = r'\+'~~          t_LT = r'\<'
~~t_MINUS = r'\-'~~         t_GT = '\>'
~~t_MULTIPLY = r'\*'~~      t_LTE = r'\<='
~~t_DIVIDE = r'\/'~~        t_GTE = r'\>='
~~t_EQUAL = r'\='~~         t_EET = r'\=='
t ~~_ignore = r'~~          t_NET = r'\b='
                           t_ignore = r' '

def t_FLOAT(t):
    r'\d+\.\d+'
    t.value = float (t.value)
    return t
```

1

```
def t_INT(t):
    r'\d+'
    t.value = int (t.value)
    return t

def t_NAME (t):
    r'[a-zA-Z][a-zA-Z-0-9]*'
    t.type ='NAME'
    return t

def t_error (t):
    print ("Illegal Characters!")
    t.lexer.skip(1)

lexer = lex.lex()
f = open('code.c','r')
~~lexer~~ Lines = f.readlines()
for l in Lines:
    lexer.input (t)
    while True:
        tok = lexer.token()
        if not toK
            break
        Print (tok)
```

1.B

```
import sys
next = None

def P():
    sys. stdout .write('\n source: *)
    scan()
    if next= = '$':
        sys.exit (1)
    E ()
    if next == '$':
        sys. stdout .write(" , accept.")
    else
        error (1)
def E():
    T ()
    while next = = '+' :
        scan()
        T()
def T ():
    F ()
    while next = = '*' :
        scan()
        F()
def F() :
    if next . is alnum() :
        scan()
    elif next == '(' :
        scan ()
        E ()
        if next == ')' :
            scan()
        else :
            error(3)
```

3

Scanned with CamScanner

```
        else:
            error(4)

def error(n):
    print("\nError: " + str(n)+"\n")
    sys.exit(1)

def getch():
    c= sys.stdin.read(1)
    if len(c) > 0:
        print (c)
        return (c)
    else:
        return None

def scan():
    global next
    next =getch()
    if next == None:
        sys.exit(1)
    while next.isspace():
        next = getch()

while True:
    P()
```

Test case

a+b*c$ —— Accept

a*b*c$ —— Accept

d+b*c*d$ —— Accept

(a+b)*c$ —— Accept

a*(b+c)$ —— Accept

(a*b*(c+d)+e)+f$ —— Accept

((a*b*(c+(d))+e)+i)$ — Accept

$

Error entries

a+b*$ —— error 4

b*a$ —— error 4

*a+b$ — error 4

a+*b$ —— error 4

~~a+*b$~~

(a+b$ — error 3

(a+b)+c)*d)$ — error 1

1.c

Servener - code.py

```
import socket
HOST =" 172.16.57.96"
PORT = 52564
with socket.socket(socket.AF_INET,socket.SOCK_STREAM)
            as s:
    s.bind((HOST, PORT))
    s. listen()
    conn, addr = s.accept()
     with conn:
         print('connected by', addr)
         while True:
             data = conn.recv(1024)
             if not data:
                 break
             conn. sendall (data)
```

client_code.py

```
import socket
HOST = '172.16.57.96'
PORT = 52564
with socket.socket(socket.AF-INET,socket.SOCK_STREAM) as s:
        s.connect ([HOST, PORT])
        s. sendall (b'hello, world')
        data = s.recv (1024)
print ('Received', repr (data))
```

$\textcircled{6}$