PASS 1 ASSEMBLER

code = []

MOT\_ref={

'L': ['L','58','4','RX'],

'A': ['A','5A','4','RX'],

'ST': ['ST','50','4','RX'],

'BASR':['BASR','0D','4','RX'],

'BALR':['BALR','05','2','RR'],

}

POT\_ref={

'START': ['START','OPCODE FOR START'],

'USING': ['USING','OPCODE FOR USING'],

'END': ['END', 'OPCODE FOR END'],

'DC': ['DC','OPCODE FOR DC'],

'DS': ['DS', 'OPCODE FOR DS']

}

lc = 0 length = 0

MOT = [] POT = [] symbolTable = [] operands = [] *# take input form file* with open('./textfile.txt') as f: code = f.readlines()

for i,line in enumerate(code): tokens = line.split(' ') print(f"LINE {i+1}:{tokens}") for token in tokens: if token in MOT\_ref:

length = int(MOT\_ref[token][2]) lc+=length

MOT.append(MOT\_ref[token]) elif token in POT\_ref:

if token == "DC" or token == 'DS': lc+=4

POT.append(POT\_ref[token]) else:

flag = True for char in token:

if char == ',' or char in '0123456789':

flag = False break if flag: symbolTable.append(tuple((token,lc,length,'R')))

else:

operands.append(tuple((token[:-1].split(','),f"line: {i+1}")))

print('\n\nMOT:') print("mnemonic\tbinary\_op\tins\_length\tins\_format")

for x in MOT:

# print(f"{x[0]}\t\t{x[1]}\t\t{x[2]}\t\t{x[3]}")

print('\n\nPOT:') print("mnemonic\topcode") for x in POT:

# print(f"{x[0]}\t\t{x[1]}")

print('\n\nSymbols:') print('symbol\tvalue\tlength\trelocation') for x in symbolTable:

# print(f"{x[0]}\t{x[1]}\t{x[2]}\t{x[3]}")

print('\n\nOperands:',operands) print('')

# textfile.txt

JOHN START 0

USING \*,15

L 1,FOUR

A 1,FIVE

ST 1,TEMP

FOUR DC F'4

FIVE DC F'5

TEMP DS 1F

END

**Output:**

