Assignment 2: Write a program to implement a two pass assembler.

Name: Adnan Sadar

Class: TY IT

Roll No: 2

Batch: 3

Code:

POT = {"Pseudo": ["Db", "Dw", "Org", "ENDP", "Const", "End"],

"No\_operands": [2, 2, 1, 1, 1, 0], "Length": [1, 1, 1, 1, 1, 1]}

MOT = {"Mnemonic": ["Add", "Sub", "Mult", "Jmp", "Jneg", "Jpos", "Jz", "Load", "Store", "Read", "Write", "Stop"], "Opcode": [

1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12], "No\_operands": [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0], "Length": [2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1]}

ST = {"Symbol": [], "Address": []}

fr = open("input.txt", "r")

lc = 0

temp = 0 # to indicate end of code

line = fr.readline()

while line:

words = line.split()

for word in words:

# Check if word is in MOT or no

for var in MOT["Mnemonic"]:

if var == word:

index = MOT["Mnemonic"].index(word)

lc += MOT["Length"][index]

flag = 0

break

if flag == 0: # Word is found in Mnemonic

flag = 1

continue

elif flag == 1 and word == "Loop:" or flag == 1 and word == "Outer:": # if word is a label

temp2 = 1 # set presence of word in ST or not as 1(false)

for var in ST["Symbol"]:

if var == word: # Check if word is already present in symbol table

temp2 = 0

break

if temp2 == 1: # if not present then add it

ST["Symbol"].append(word)

idx = ST["Symbol"].index(word)

# set address of the symbol just appended

ST["Address"].append(lc)

elif flag == 1 and word != "Endp" and word != "Outer" and word != "Loop": # to distinguish from labels

temp1 = 1

for var in ST["Symbol"]:

if var == word: # checking if word is already present in symbol table

temp1 = 0

break

if temp1 == 1:

ST["Symbol"].append(word)

ST["Address"].append(0)

elif word == "Endp":

temp = 1 # to indicate end of code

address = lc

print("Symbol Table After First Pass: \n")

print(ST)

break

if temp == 1:

line = fr.readline()

while line:

words = line.split()

for word in words:

for var in ST["Symbol"]:

if word == var:

temp3 = 0

index = ST["Symbol"].index(word)

break

else:

temp3 = 1 # word not found in ST

if temp3 == 0:

continue # The Word has been found in Mnemonic and we want to find length of next word in POT

else:

for var in POT["Pseudo"]:

if word == var:

idx = POT["Pseudo"].index(word)

# ST address is updated in 2nd pass

ST["Address"][index] = address

# increment the address acc to length

address += POT["Length"][idx]

line = fr.readline()

print("Symbol Table After Second Pass: \n")

print(ST)

break

else:

line = fr.readline()

fr.close()

print("\n")

print("Output Program: \n")

fr = open("input.txt", "r")

line = fr.readline()

temp1 = 0

while line:

words = line.split()

for word in words:

if word == "Endp":

temp1 = 1

break

for var in MOT["Mnemonic"]: # check if word is present in Mnemonic

if var == word:

flag = 1

index = MOT["Mnemonic"].index(word)

print(MOT["Opcode"][index], *end*=' ')

break

if flag == 1:

flag = 0 # word is found in Mnemonic and read next word

continue

elif flag == 0:

for var in ST["Symbol"]:

if word == var or word == "Outer" or word == "Loop":

if word == "Outer":

word = word.replace("Outer", "Outer:")

elif word == "Loop":

word = word.replace("Loop", "Loop:")

elif word == "Outer:" or word == "Loop:":

continue

idx = ST["Symbol"].index(word)

print(ST["Address"][idx])

break

if temp1 == 1:

break

print("\n")

line = fr.readline()

Output:



