

## Program and Output

### Program :

```
# -*- coding: utf-8 -*-  
"""
```

Created on Wed Mar 9 07:40:28 2022

@author: SANKETHARVANDE.CS

Title: Implementation of two pass assembler

This program will work properly for certain set of assembly commands only  
"""

```
fp=open('program.txt','r')  
program=fp.read().split("\n")
```

```
#print(program)
```

```
fp.close()
```

```
mnemonic_tab={'STOP':'00','ADD':'01','SUB':'02','MULT':'03','MOVER':'04','MOVEM':'05','COMP':'06','BC':'07','DIV':'08','READ':'09','PRINT':'10','DS':'01','DC':'02'}
```

```
reg_code={'AREG':1,'BREG':2,'CREG':3,'DREG':4}
```

```
condition_code={'LT':1,'LE':2,'EQ':3,'GT':4,'GE':5,'ANY':6}
```

```
optab={'STOP':'AD','ADD':'IS','MULT':'IS','MOVER':'IS','MOVEM':'IS','COMP':'IS','DC':'DL','DS':'DL','READ':'IS','PRINT':'IS','BC':'IS'} # declaring operands
```

and their respective types of sentences Imp sent, Decl sent and AD

```
sym_table={} #empty symbol table
```

```
print('Content of Mnemonic Table is :\n')
```

```
print('Mnemonic',' Code\n')
```

```
for k,v in mnemonic_tab.items():
```

```
    print('{0}    {1}'.format(k, v))
```

```
print()
```

```
print('Content of Opcode table is:\n')
```

```
print('Mnemonic','Class\n')
```

```

for k,v in optab.items():
    print('{0}    {1}'.format(k,v))
print()
print('Input Assembly Code')
print()

```

```

#print the source code
#set the value of lc
"""

```

```

START 200
MOVER AREG FIRST
ADD AREG SECOND
MOVEM AREG RESULT
PRINT RESULT
RESULT DS 1
FIRST DC 5
SECOND DC 7
END
"""

```

```

for line in program: #traversing line by line through the program
    a=line.split() # splitting the line and it will get converted into list of strings
    if a[0]=='START': # checking the first element in the list if it is a start symbol
        and if it is not simply print the line.
        lc=int(a[1]) # converting the 1st element in the list into integer value and
        storing it into lc
        temp=lc # the int value stored in lc is stored in temp (here storing the
        address value as a temporary value)
        print(line) # printing the line

```

```

#Build the symbol table

```

```

for line in program:
    l=line.split() # splitting the lines into list of single words and storing it into l
    for i in l: # traversing through the l one by one
        if i not in optab and i not in reg_code and i.isdigit()!=True and i not in
        condition_code: # START FIRST SECOND RESULT END
            sym_table[i]=lc
            lc+=1
print()
print('Content of Symbol Table is:')

```

```

print()
print('Symbol Name','Address')
for k,v in sym_table.items():
    print('{0}      {1}'.format(k,v))

lc=temp # taking the value from temp into lc (lc = 200)
print()
print('Intermediate code after PASS-1')
print()
a=list(sym_table.keys()) # making a list of all the keys from symbol table and
storing them into a
for line in program: # traversing line by line through assembly program
    lexeme=line.split() # Splitting the assembly code line by line and storing it in
the list in the variable lexeme
    if(len(lexeme)==4): # if lexeme is greater than length 4 i.e. there are 4
elements in the list then remove 1st lexeme it from the list
        lexeme.remove(lexeme[0])
        if lexeme[0] in optab:
            if optab[lexeme[0]]=='AD':
                if(len(lexeme)==1):

print(lc,(optab[lexeme[0]],mnemonic_tab[lexeme[0]]),(C,',lexeme[0],'))

        if lexeme[0] in optab:
            if optab[lexeme[0]]=='IS':
                if len(lexeme)==3:
                    if lexeme[0]=='BC':

print(lc,(optab[lexeme[0]],mnemonic_tab[lexeme[0]]),condition_code[lexeme[1]
],('S',a.index(lexeme[2]),'))
                lc+=1
            else:

print(lc,(optab[lexeme[0]],mnemonic_tab[lexeme[0]]),reg_code[lexeme[1]],('S',
a.index(lexeme[2]),'))
                lc+=1
            if(len(lexeme)==2):

print(lc,(optab[lexeme[0]],mnemonic_tab[lexeme[0]]),(S',a.index(lexeme[1]),'))
                lc+=1

```

```

        if lexeme[0] not in optab:
            if len(lexeme)==3:

print(lc,(optab[lexeme[1]],mnemonic_tab[lexeme[1]]),(C',lexeme[2],'))
        lc+=1

        if len(lexeme)==4:
            print(lc,(optab[lexeme[1]],mnemonic_tab[lexeme[1]]),)
            lc+=1
print()
print('Machine Code after PASS II \n')

lc=temp # again we are reseting the lc as a starting address of the program
here 200

for line in program:
    lexeme=line.split() # making a list line by line and storing it into lexeme
    if len(lexeme)==4: # 4 elements in the list then remove the first element
        lexeme.remove(lexeme[0])

    if lexeme[0]in optab:
        if optab[lexeme[0]]=='AD':
            if(len(lexeme)==1):
                print()
                lc+=1
            else:
                if(lexeme[0]=='START'):pass
    if lexeme[0] in optab:
        if optab[lexeme[0]]=='IS':
            if len(lexeme)==3: # if there are 3 elements in the list
                if lexeme[0]=='BC':

print(lc,mnemonic_tab[lexeme[0]],condition_code[lexeme[1]],sym_table[lexem
e[2]])
        lc+=1
    else:

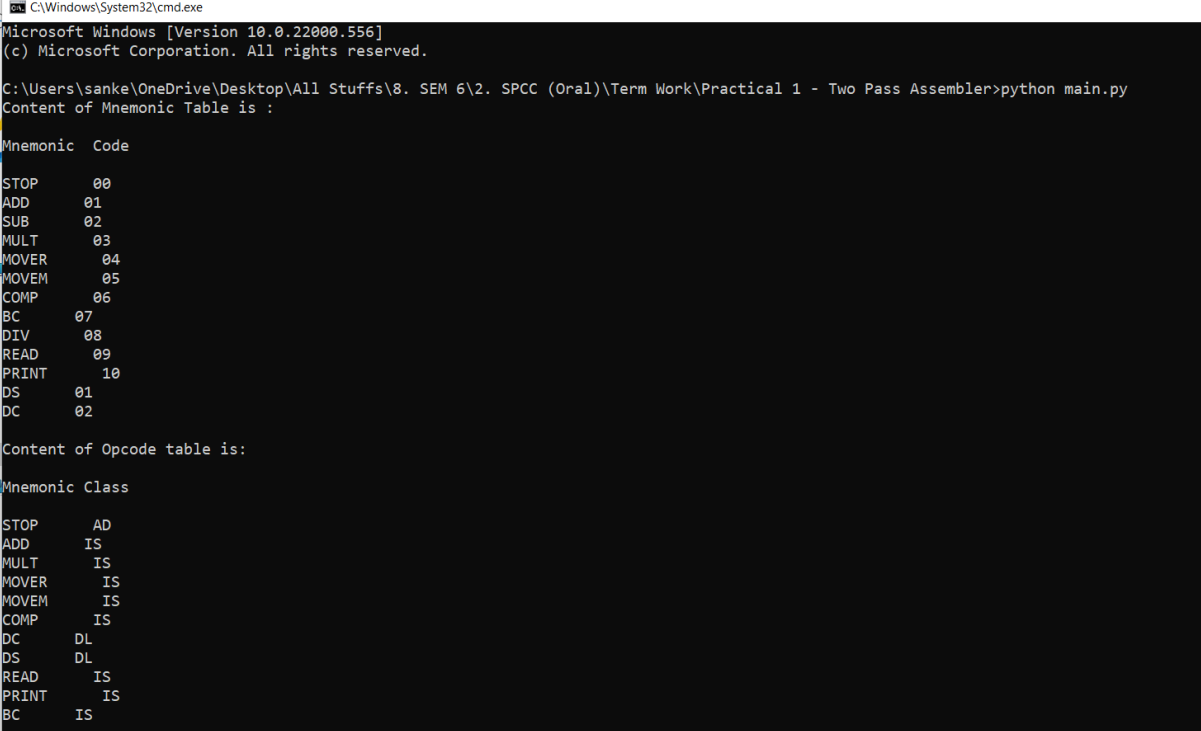
print(lc,mnemonic_tab[lexeme[0]],reg_code[lexeme[1]],sym_table[lexeme[2]])
        lc+=1
        if(len(lexeme)==2):

```

```
print(lc,mnemonic_tab[lexeme[0]],sym_table[lexeme[1]])  
lc+=1
```

```
if lexeme[0] not in optab:  
    if len(lexeme)==3:  
        print(lc,mnemonic_tab[lexeme[1]],lexeme[2])  
        lc+=1  
    if len(lexeme)==4:  
        print(mnemonic_tab[lexeme[1]])  
        lc+=1
```

## Output :



```
C:\Windows\System32\cmd.exe  
Microsoft Windows [Version 10.0.22000.556]  
(c) Microsoft Corporation. All rights reserved.  
  
C:\Users\sanke\OneDrive\Desktop\All Stuffs\8. SEM 6\2. SPCC (Oral)\Term Work\Practical 1 - Two Pass Assembler>python main.py  
Content of Mnemonic Table is :  
  
Mnemonic  Code  
-----  
STOP      00  
ADD        01  
SUB        02  
MULT       03  
MOVER      04  
MOVEM      05  
COMP       06  
BC         07  
DIV        08  
READ       09  
PRINT      10  
DS         01  
DC         02  
  
Content of Opcode table is:  
  
Mnemonic Class  
-----  
STOP      AD  
ADD        IS  
MULT       IS  
MOVER      IS  
MOVEM      IS  
COMP       IS  
DC         DL  
DS         DL  
READ       IS  
PRINT      IS  
BC         IS
```

Select C:\Windows\System32\cmd.exe

Input Assembly Code

```
START 200
MOVER AREG FIRST
ADD AREG SECOND
MOVEM AREG RESULT
PRINT RESULT
RESULT DS 1
FIRST DC 5
SECOND DC 7
END
```

Content of Symbol Table is:

Symbol Name	Address
START	200
FIRST	206
SECOND	207
RESULT	205
END	208

Intermediate code after PASS-1

```
200 ('IS', '04') 1 (S 1 )
201 ('IS', '01') 1 (S 2 )
202 ('IS', '05') 1 (S 3 )
```

Machine Code after PASS II

```
200 04 1 206
201 01 1 207
202 05 1 205
203 10 205
204 01 1
205 02 5
206 02 7
```

C:\Users\sanke\OneDrive\Desktop\All Stuffs\8. SEM 6\2. SPCC (Oral)\Term Work\Practical 1 - Two Pass Assembler>