Experiment - 10 WAP to display pseudo Opcode & Location counter for given assembly level code

Name : **Ansari Mohd Adnan** **Rollno :** **17DCO63**
Class : **TE.CO** **Batch :** **B3**

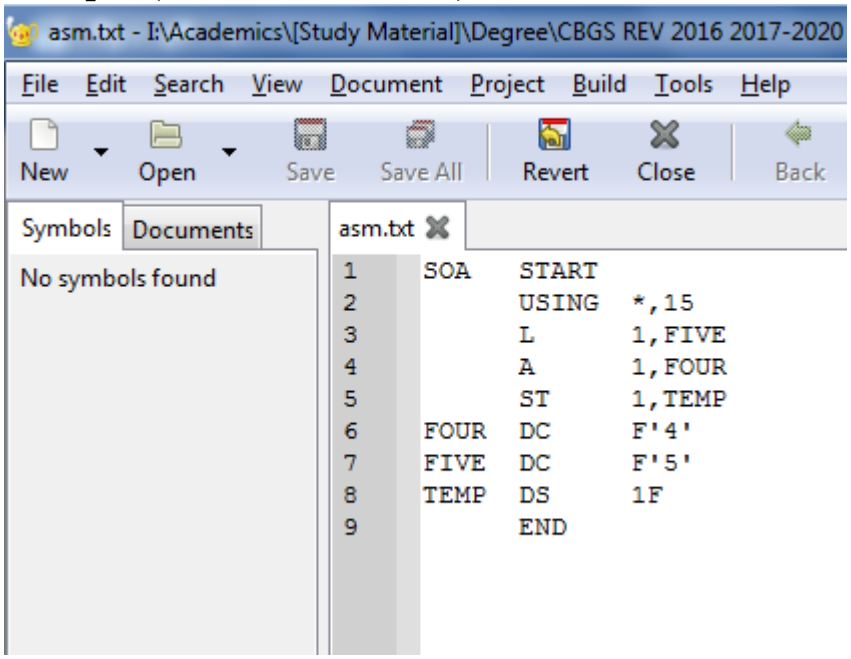
#Source Code in python

```
import re
from prettytable import PrettyTable
t=PrettyTable(['SOURCE PROGRAM','First Pass','Second Pass'])
s_index=0
index=[]
index2=[]
newdata=[]
row2=[]
row3=[]
fr=open("asm.txt","r")
data=fr.readlines()

for i in range(len(data)):
        if 'START' in data[i]:
                s_index=re.search("START+\s+[0-9]+",data[i])
                if(s_index):
                        s_index=s_index.group(0)
                        s_index=re.search("[0-9]+",s_index)
                        s_index=s_index.group(0)
                        index.append(s_index)
                        newdata.append(data[i])
                        row2.append(str(s_index)+"  ")
                        i+=1
                else:
                        s_index=0
                        index.append(s_index)
                        newdata.append(data[i])
                        row2.append(str(s_index)+"  ")
                        i+=1
                if 'USING' in data[i]:
                        index.append(s_index)
                        row2.append(str(s_index)+"  ")
                        newdata.append(data[i]+"  ")
                        i+=1
                        line=re.search("[A-Z]+\s+[0-9]+,+[A-Z]+",data[i])
                        var=line.group(0)
                        s_var=re.search("[A-Z]{1,}",var)
```

```
            s_var=s_var.group(0)
            index.append(s_index)
            newdata.append(data[i])
            row2.append(str(s_index)+str("  "+s_var))
            i+=1
            j=i
    for j in range(i,len(data)):
            try:
                    line=re.search("[A-Z]+\s+[0-9]+,+[A-Z]+",data[j])
                    line2=re.search("[A-Z]+\s+([DC]+|[DS])+\s+(F+'[0-9]'+|[0-
                    9]+F)+",data[j])
                    if(line):
                            var=line.group(0)
                            s_var=re.search("[A-Z]{1,}",var)
                            s_var=s_var.group(0)
                            s_index+=4
                            index.append(s_index)
                            newdata.append(data[j])
                            row2.append(str(s_index)+str("  "+s_var))
                            j+=1
                    elif(line2):
                            get=line2.group(0)
                            getvalue=re.search("F+'[0-9]'",get)
                            if(getvalue):
                                    getvalue=getvalue.group(0)
                                    getvalue=re.search("[0-9]",getvalue)
                                    getvalue=getvalue.group(0)
                                    getvalue=bin(int(getvalue))
                                    s_index+=4
                                    index.append(s_index)
                                    newdata.append(data[j])
                                    row2.append(str(s_index)+str("
                                    ")+str(getvalue).lstrip('0b'))
                            else:
                                    s_index+=4
                                    index.append(s_index)
                                    newdata.append(data[j])
                                    row2.append(str(s_index))

                                    j+=1
                    else:
                            s_index+=4
                            index.append(s_index)
                            newdata.append(data[j])
                            row2.append(str(s_index))
```

```
                                        j+=1
                        except:
                                pass
for i in range(len(newdata)):
        if 'USING' in newdata[i]:
                usval=re.search("[0-9]+",newdata[i])
                usval=usval.group(0)
        line3=re.search("[A-Z]+\s+[0-9]+,+[A-Z]+",newdata[i])
        if(line3):
                line3=line3.group(0)
                s_var=re.search("[A-Z]{1,}",line3)
                s_var=s_var.group(0)
                var_label=re.search("[A-Z]{1,}$",line3)
                var_label=var_label.group(0)
                for j in range(len(newdata)):
                        line4=re.search("[A-Z]+\s+([DC]+|[DS])+\s+(F+'[0-9]'+|[0-
                        9]+F)+",newdata[j])
                        if(line4):
                                line4=line4.group(0)
                                var_label2=re.search("[A-Z]{1,}",line4)
                                var_label2=var_label2.group(0)
                                if(var_label==var_label2):
                                        row3.append(str(index[i])+str("  "+s_var+"
                                        ")+str("1,")+str(index[j])+str("(0,"+usval+")"))

        else:
                getvalue=re.search("F+'[0-9]'",newdata[i])
                if(getvalue):
                        newgetvalue=getvalue.group(0)
                        if newgetvalue in newdata[i]:
                                getvalue=re.search("[0-9]",newgetvalue)
                                getvalue=getvalue.group(0)
                                getvalue=bin(int(getvalue))
                                row3.append(str(index[i])+str(" ")+str(getvalue).lstrip('0b'))
                                i=+1
                else:
                        row3.append(index[i])
for i in range(len(data)):
        if 'END' in data[i]:
                t.add_row([data[i],index[i],index[i]])
        else:
                t.add_row([newdata[i],row2[i],row3[i]])

t.align='c'
print("\n")
print(t)
```

#Output (Content of asm.txt)



#Output