Ahsanullah University of Science and Technology (AUST)
Department of Computer Science and Engineering

Assignment-3

Course No.: CSE4130

Course Title: Formal Languages & Compilers Lab

Date of Submission-
12/06/2023

Submitted To-
Submitted To- Mr. Md. Aminur Rahman & Iffatur Nessa.

Submitted By-
MD Fardin Jaman Aranyak
190204093
Group: B2
Year- 4th
Semester- 1st
Session: Fall’22
Department- CSE

```
from tabulate import tabulate


#variable

lexemes=""

copy_lexemes=""

tokenToBeRemove=["kw","op","num","sep","par","brc"]

dataType=["double","int","float"]

id_names_withDataType=[]

id_names_withType=[]

id_names_withValue=[]

symbol_table=[]


#read file

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

lexemes=file.read()


#create space between [ ]

for i in range(len(lexemes)):

    if lexemes[i]=="[":

        copy_lexemes+=lexemes[i]+" "

    elif lexemes[i]=="]":

         copy_lexemes+=" "+lexemes[i]

    else:

        copy_lexemes+=lexemes[i]


#print(copy_lexemes)

#print()


#seperate every keyword
```

```
lexemes_list=copy_lexemes.split()


#only identifiers are kept
for items in  lexemes_list:
    if(items in tokenToBeRemove):
        lexemes_list.remove(items)


#print(lexemes_list)
#print()


scope_flag=0
for i in range(len(lexemes_list)):
    if(lexemes_list[i]=="id" and lexemes_list[i+4]=="("):
        scope=lexemes_list[i+1]
        scope_flag=1;
    elif(lexemes_list[i]=="id" and lexemes_list[i+1]=="main" and lexemes_list[i+4]=="("):
        scope="main"
        scope_flag=1;
    elif(lexemes_list[i]=="}"):
        scope_flag==0
    elif(scope_flag==0):
        scope="global"
    if(lexemes_list[i]=="id" and lexemes_list[i-3] in dataType):
        #print(lexemes_list[i+1]," ",lexemes_list[i-3])
        if(lexemes_list[i+1]=="main"):
            id_names_withDataType.append(["global",lexemes_list[i+1],lexemes_list[i-3]])
        else:
            id_names_withDataType.append([scope,lexemes_list[i+1],lexemes_list[i-3]])
        if(lexemes_list[i+4]=="("):
            id_names_withType.append([scope,lexemes_list[i+1],"func"])
        else:
            id_names_withType.append([scope,lexemes_list[i+1],"var"])
        #if(lexemes_list[i+4]=="="):
           # id_names_withValue.append([lexemes_list[i+1],lexemes_list[i+7]])

scope_flag=0
for i in range(len(lexemes_list)):
    if(lexemes_list[i]=="id" and lexemes_list[i+4]=="("):
        scope=lexemes_list[i+1]
        scope_flag=1;
    elif(lexemes_list[i]=="id" and lexemes_list[i+1]=="main" and lexemes_list[i+4]=="("):
        scope="main"
        scope_flag=1;
    elif(lexemes_list[i]=="}"):
        scope_flag==0
    elif(scope_flag==0):
        scope="global"
    if(lexemes_list[i]=="id"):
        if(lexemes_list[i+4]=="=" and lexemes_list[i+7]!='id'):
            id_names_withValue.append([scope,lexemes_list[i+1],lexemes_list[i+7]])
#print(id_names_withValue)
#print(id_names_withType)
#print(id_names_withDataType)

sn=0
for i in range(len(id_names_withDataType)):
    sn+=1
    name=id_names_withDataType[i][1];
    idType=id_names_withType[i][2];
    dtType=id_names_withDataType[i][2];
    scp=id_names_withDataType[i][0];
    values="\0"
    for j in range(len(id_names_withValue)):
        if(name==id_names_withValue[j][1] and scp==id_names_withValue[j][0]):
                values=id_names_withValue[j][2];
    symbol_table.append([sn,name,idType,dtType,scp,values])
#print()
#print(symbol_table)
file.close()




def display():
    if not symbol_table:
        print("Symbol table is empty.")
    else:
        data = [
            ["Sl. No.", "Name", "ID Type", "Data Type", "Scope", "Value"],
        ]
        for i in range(len(symbol_table)):
            data.append(symbol_table[i])
        print(tabulate(data, headers="firstrow", tablefmt="grid"))


def lookup():
    name = input("Enter an Identifier's Name: ")
    data = [
        ["Sl. No.", "Name", "ID Type", "Data Type", "Scope", "Value"],
    ]
    flag = 0
    for entry in symbol_table:
        if name == entry[1]:
            data.append(entry)
            flag = 1
    if flag == 1:
        print(tabulate(data, headers="firstrow", tablefmt="grid"))
    else:
        print("Data Not Found!!")

def free():
    symbol_table.clear()
    print("Symbol table has been cleared.")

def set_attribute():
    name, scope = input("Enter the variable Name and Scope to Update Value: ").split()
    value = input("Enter Value: ")
    for entry in symbol_table:
        if entry[1] == name and entry[4] == scope:
            entry[5] = value
            print("Attribute updated successfully.")
            return
    print("Variable not found in the symbol table.")

def insert():
    name, idType, dataType, scope, value = input("Enter Name, ID-Type, Data-Type, Scope, Value: ").split()
    symbol_table.append([len(symbol_table) + 1, name, idType, dataType, scope, value])
    print("Entry added to the symbol table.")
```

```
while True:
    print("\nA. Insert\nB. Set Attribute\nC. Free\nD. Look Up\nE. Display\n")
    mode = input("Enter the mode (A, B, C, D, or E): ")

    # Process the user's choice
    if mode == "A":
        insert()
    elif mode == "B":
        set_attribute()
    elif mode == "C":
        free()
    elif mode == "D":
        lookup()
    elif mode == "E":
        display()
    else:
        print("Invalid mode selection.")
```