AHSANULLAH UNIVERSITY OF SCIENCE AND TECHNOLOGY
DHAKA-1208, BANGLADESH.

Department of Computer Science and Engineering
Spring 2019

Program: Bachelor of Science in Computer Science and Engineering

Course No: CSE 4130

Course Title: Formal Languages and Compilers Lab

Session: Spring 2019

Assignment No: 03

Date of Submission: 04/09/2019

Submitted to Md. Aminur Rahman

Lecturer, Department of CSE, AUST.

Shoeb Mohammad Shahriar
Lecturer, Department of CSE, AUST.

Submitted By

Name : Robiul Hasan Nowshad
Student ID : 16.01.04.061
Lab Group : B1

I. Question

Suppose, a given C source program has been scanned, filtered and then lexically analysed as it was done in Session 1 & 2. We have all the lexemes marked as different types of tokens like keywords, identifiers, operators, separators, parentheses, numbers, etc. We get corrected the unknown lexemes first, and then generate a Symbol Table describing the features of the identifiers. Finally, we generate a modified token stream in accordance with the Symbol Table for processing by the next phase, that is, Syntactic Analysis.

II. Input File

input.txt

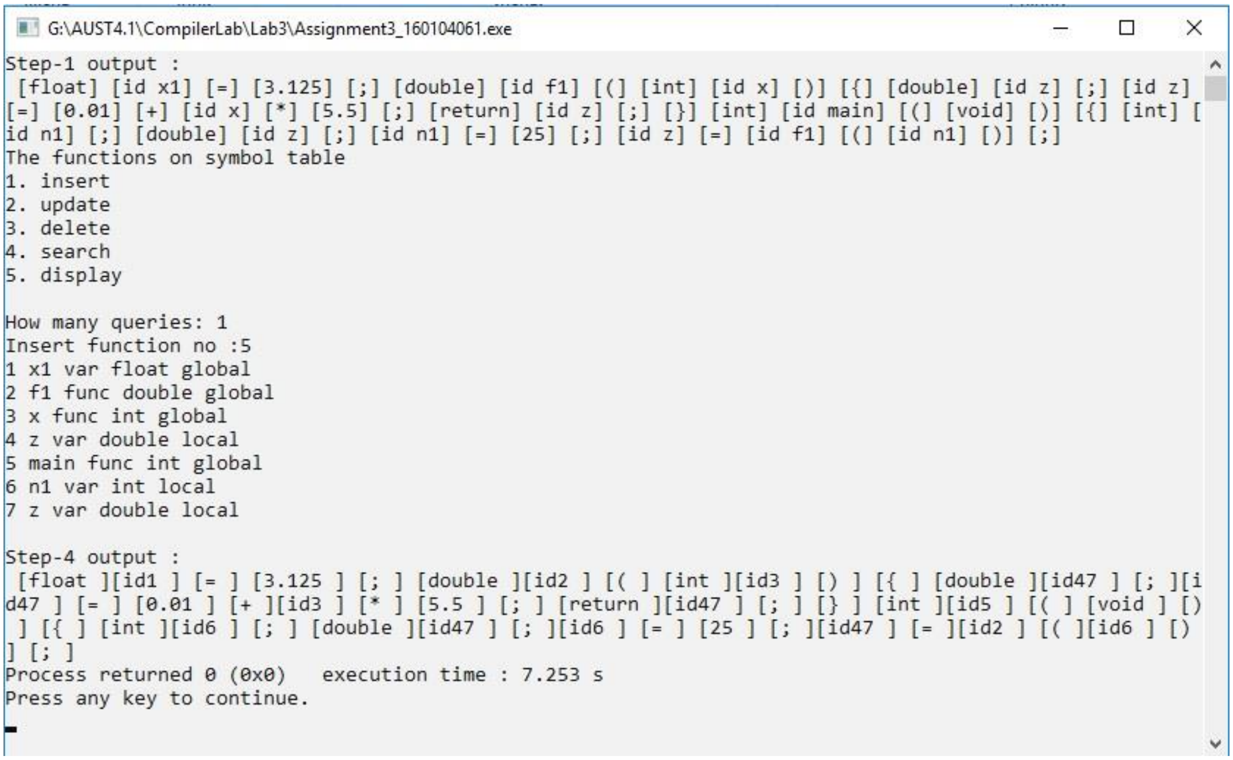
kw float] [id x1] [op =] [num 3.125] [sep ;] [kw double] [id f1] [par (] [kw int] [id x] [par)] [brc {] [kw double] [id z] [sep ;] [id z] [op =] [num 0.01] [op +] [id x] [op *] [num 5.5] [sep ;] [kw return] [id z] [sep ;] [brc }] [kw int] [id main] [par (] [kw void] [par)] [brc {] [kw int] [id n1] [sep ;] [kw double] [id z] [sep ;] [id n1] [op =] [num 25] [sep ;] [id z] [op =] [id f1] [par (] [id n1] [par)] [sep ;]

III. Source Code

Assmnt3_160104061.c

```
 #include<bits/stdc++.h>
using namespace std;

int step1();
int step2();
int step3();
int step4();
struct node{
string name,type,data,scope;
};
node tmp;
vector<node>v;
stack<char>sb;
FILE *f1,*f2,*f3;
string s;

int main(){

    step1();
    step2();
    step3();
    step4();
}

int step1(){
    f1 = fopen("input.txt", "r");
    f2 = fopen("output1.txt", "w");
    char str[50];
    while(fscanf(f1, "%s", &str)!=EOF)
    {
        if((!strcmp(str,"[kw"))||(!strcmp(str,"[op"))||(!strcmp(str,"[num"))||(!strcmp(str,"[sep"))||(!strcmp(str,"[par"))||(!strcmp(str,"[brc"))||(!strcmp(str,"[kw"))){
            fprintf(f2, " [");
            fscanf(f1, "%s", &str);
            fprintf(f2, "%s", str);
        }
        else fprintf(f2, " %s", str);
    }
    fclose(f1);
    fclose(f2);

    printf("Step-1 output :\n");
    f1 = fopen("output1.txt", "r");
    char c;
    while((c=fgetc(f1))!=EOF)
    {
        printf("%c",c);
    }
    fclose(f1);
    printf("\n");
}


int step2(){
    string filename="input.txt";
    ifstream fin( filename.c_str() );


    getline( fin, s);

    for(int i = 1;i<s.size()-1;i++)
    {
        if(s[i]=='{'){
        sb.push(s[i]);
        }
        if(s[i]=='}'){
            sb.pop();
        }

        if(s[i-1]=='['&&s[i]=='i'&&s[i+1]=='d'){
            int j = i-4;
            i = i+3;
            string ret = "";
            while(s[i]!=']'){
                ret+=s[i];
                i++;
            }
            string st = "";
            while(s[j]!=' '){
                st+=s[j];
                j--;
            }
            reverse(st.begin(),st.end());
            if(st=="double"||st=="int"||st=="float") {}
            else continue;
            if(s[i+3]=='p'){
                tmp.name = ret;
                tmp.type = "func";
                tmp.data = st;
                tmp.scope = "global";
            }
            else{
                tmp.name = ret;
```

```
         tmp.type = "var";
         tmp.data = st;
     if(!sb.empty()) tmp.scope = "local";
     else tmp.scope = "global";
     }

     v.push_back(tmp);
        }
    }

}
int step3(){
    printf("The functions on symbol table\n1. insert\n2. update\n3. delete\n4. search\n5. display\n\nHow many queries: ");
    int n;
    int idx = 1;
    cin>>n;
    while(n--)
    {
        printf("Insert function no :");
        string name,type,data,scope;
        int fn;
        cin>>fn;

        if(fn==1){
            cin>>name>>type>>data>>scope;
            tmp.name = name;
            tmp.type = type;
            tmp.data = data;
            tmp.scope = scope;
            v.push_back(tmp);
        }
        else if(fn==2){
            cin>>idx>>name>>type>>data>>scope;
            for(int k = 1;k<=v.size();k++){
                if(idx==k){
                    tmp.name = name;
                    tmp.type = type;
                    tmp.data = data;
                    tmp.scope = scope;
                    v[k-1] = tmp;
                }
            }
        }
        else if(fn==3){
            cin>>idx;
            v.erase(v.begin()+idx-1);
        }
        else if(fn==4){
            cin>>idx;
            for(int k = 1;k<=v.size();k++){
                if(idx==k){
                    cout<<idx<<" "<<v[k-1].name<<" "<<v[k-1].type<<" "<<v[k-1].data<<" "<<v[k-1].scope<<endl;
                }
            }
        }
        else if(fn==5){
            for(int k = 1;k<=v.size();k++)
            {
                cout<<k<<" "<<v[k-1].name<<" "<<v[k-1].type<<" "<<v[k-1].data<<" "<<v[k-1].scope<<endl;
            }
        }
    }
}
int step4(){
    f1 = fopen("output1.txt", "r");
    f2 = fopen("output2.txt", "w");

    char c;
    while((c=fgetc(f1))!=EOF)
    {
        if(c==']')
            fprintf(f2," ]");
        else putc(c,f2);
    }
    fclose(f1);
    fclose(f2);
    printf("\n");

    f1 = fopen("output2.txt", "r");
    f2 = fopen("output3.txt", "w");

    char str[50];
    while(fscanf(f1, "%s", &str)!=EOF)
    {
        if(!strcmp(str,"[id")){
            fprintf(f2, "%s", str);
            fscanf(f1, "%s", &str);
            for(int k = 1;k<=v.size();k++)
            {
                if(str==v[k-1].name){
                    char st[10]={0};
                    itoa(k,st,10);
                    fprintf(f2,"%s", st);
                }
            }
        }
        else fprintf(f2, " %s", str);
    }
    fclose(f1);
    fclose(f2);
    printf("Step-4 output :\n");
    f1 = fopen("output3.txt", "r");
    while((c=fgetc(f1))!=EOF)
    {
        printf("%c",c);
    }
    fclose(f1);
}
```



```
Step-1 output :
 [float] [id x1] [=] [3.125] [;] [double] [id f1] [(] [int] [id x] [)] [{] [double] [id z] [;] [id z]
[=] [0.01] [+] [id x] [*] [5.5] [;] [return] [id z] [;] [}] [int] [id main] [(] [void] [)] [{] [int] [
id n1] [;] [double] [id z] [;] [id n1] [=] [25] [;] [id z] [=] [id f1] [(] [id n1] [)] [;]
The functions on symbol table
1. insert
2. update
3. delete
4. search
5. display

How many queries: 1
Insert function no :5
1 x1 var float global
2 f1 func double global
3 x func int global
4 z var double local
5 main func int global
6 n1 var int local
7 z var double local

Step-4 output :
 [float ][id1 ] [= ] [3.125 ] [; ] [double ][id2 ] [( ] [int ][id3 ] [) ] [{ ] [double ][id47 ] [; ][i
d47 ] [= ] [0.01 ] [+ ][id3 ] [* ] [5.5 ] [; ] [return ][id47 ] [; ] [} ] [int ][id5 ] [( ] [void ] [)
 ] [{ ] [int ][id6 ] [; ] [double ][id47 ] [; ][id6 ] [= ] [25 ] [; ][id47 ] [= ][id2 ] [( ][id6 ] [)
] [; ]
Process returned 0 (0x0)   execution time : 7.253 s
Press any key to continue.
```