DESIGN COMPILER LAB WEEK 2

Name-Lehar sancheti
Roll no-Ap19110010448
Date-12-08-2021

Program 1.

Implement lexical analyser using C for recognizing the following tokens:

• 24 keywords (given in the following program)
• Identifiers with the regular expression : letter(letter | digit)*
• Integers with the regular expression: digit+
• Relational operators: <, >, <=, >=, ==, !=
• Ignores everything between multi line comments (/* .... */)
• Storing identifiers in symbol table.

C CODE

```
#include<stdio.h>
#include<ctype.h>
#include<string.h>
#include<stdlib.h>
#define SIZE 20
void display();
struct DataItem {
   char data[20];
   int key;
   char type[15];
};
struct DataItem* hashArray[SIZE];
struct DataItem* obj1;
struct DataItem* obj2;

int hashCode(int key) {
   return key % SIZE;
}
```

```
void insert(int key,char *data,char *type) {

   struct DataItem *obj2 = (struct DataItem*) malloc(sizeof(struct DataItem));
   strcpy(obj2->data,data);
   obj2->key = key;
   strcpy(obj2->type,type);
   int hashIndex = hashCode(key);
    while(hashArray[hashIndex] != NULL && hashArray[hashIndex]->key != -1) {
     ++hashIndex;
               hashIndex %= SIZE;
   }

   hashArray[hashIndex] = obj2;
}

char keyword[30][30]={"int","while","break","for","do","if","float","char","switch",
"double","short","long","unsigned","sizeof","else","register","extern","static","auto"
,"case","break","volatile","enum","typedef","strcmp","return"};
char id[20], num[10],rel[5];

int check_keyword(char s[])
{
int i;
for(i=0;i<26;i++)
if(strcmp(s,keyword[i])==0)
return 1;
return 0;
}

int main()
{int k=0;
obj1 = (struct DataItem*) malloc(sizeof(struct DataItem));
obj1->key = -1;
FILE *fp1,*fp2;
char c;
int state=0;
int i=0,j=0,t=0;
fp1=fopen("x.txt","r");//input file containing src prog
fp2=fopen("y.txt","w");//output file name

while((c=fgetc(fp1))!=EOF)
{
switch(state)
{
case 0: if(isalpha(c)){
state=1; id[i++]=c;}
else if(isdigit(c)){
state=3; num[j++]=c;}
else if(c=='<' || c=='>'){
rel[t]=c;
state=5;
t++;
}
```

```
else if(c=='=' || c=='!')
{
rel[t]=c;
state=8;
t++;
}
else if(c=='/')
state=10;
else if(c==' ' || c=='\t' || c=='\n')
state=0;
else
fprintf(fp2,"\n%c",c);
break;
case 1:if(isalnum(c)){
state=1; id[i++]=c;
}
else{
id[i]='\0';
if(check_keyword(id)){
fprintf(fp2," \n %s : keyword ",id);
insert(k,id,"keyword");
}
else{
fprintf(fp2,"\n %s : identifier",id);
// call a function which stores id in symbol table
insert(k,id,"identifier");
}
k++;
state=0;
i=0;
ungetc(c,fp1);
}
break;
case 3:if(isdigit(c)){
num[j++]=c;
state=3;
}
else{
num[j]='\0';
fprintf(fp2," \n%s: number",num);
state=0;
j=0;
ungetc(c,fp1);

}
break;
case 5:if(c=='='){
  rel[t]=c;
  t++;
  rel[t]='\0';
fprintf(fp2,"\n%s relational operator ",rel);
t=0;

state=0;
}
else{
    rel[t]='\0';
fprintf(fp2,"\n%s relational operator ",rel);

state=0;
ungetc(c,fp1);
t=0;
}
break;
case 8:if(c=='='){
  rel[t]=c;
  t++;
  rel[t]='\0';
fprintf(fp2,"\n%s relational operator ",rel);
t=0;
state=0;
}
else{
ungetc(c,fp1);
state=0;
}
break;
case 10:if(c=='*')
state=11;
else
fprintf(fp2,"\n invalid lexeme");
break;
case 11: if(c=='*')
state=12;
else
state=11;
break;
case 12:if(c=='*')
state=12;
else if(c=='/')
state=0;
else
state=11;
break;

}//End of switch
}//end of while
if(state==11)
fprintf(fp2,"comment did not close");
fclose(fp1);
fclose(fp2);
display();
return 0;
}
void display() {
   int i = 0;
              printf("%s","SRNO\t\t\tID\t\t\ttype");
   for(i = 0; i<SIZE; i++) {
      if(hashArray[i] != NULL){
         printf("\n");
         printf(" %d%s%s%s%s",hashArray[i]->key,"\t\t\t",hashArray[i]->data,"\t\t\t",hashArray[i]->type);

      }

   }

   printf("\n");
}
```

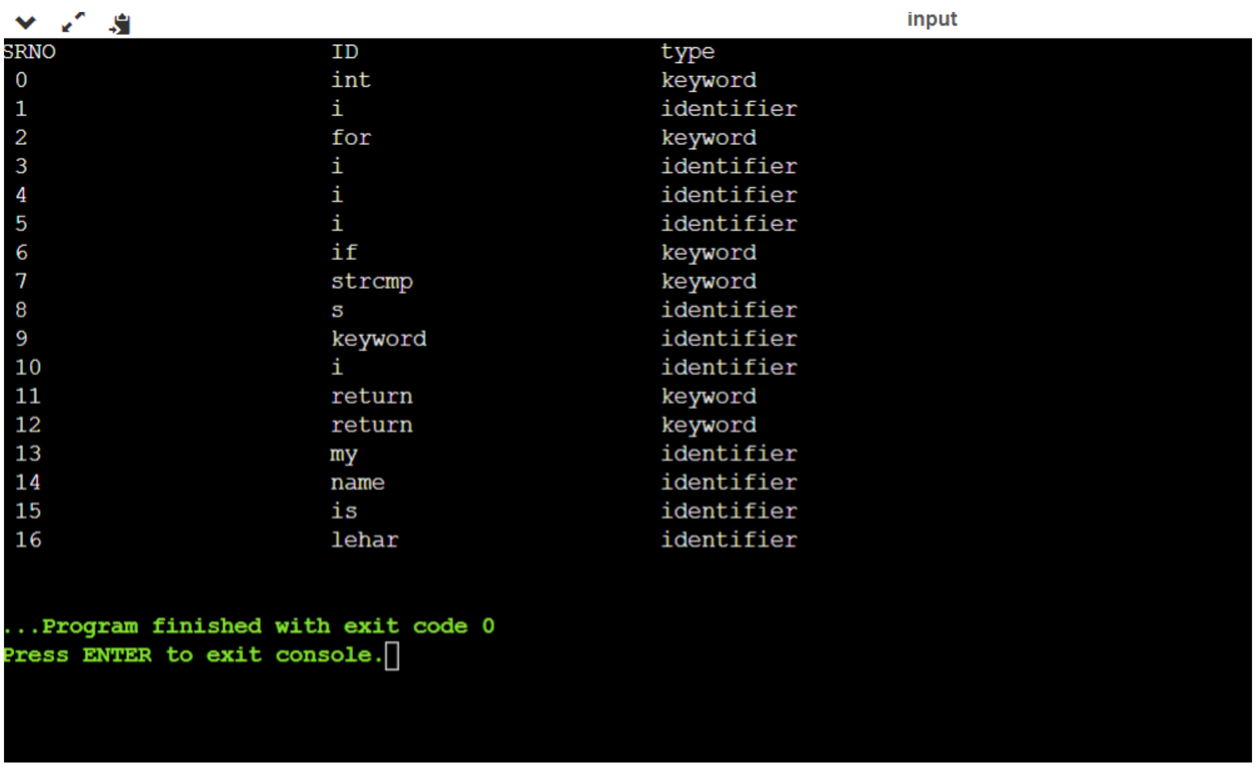
X.txt(Input file)

```
int i;
for(i=0;i<=24;i++)
if(strcmp(s,keyword[i])==0)
return 1;
return 0;
my name is lehar
```

Y.txt(Output file)

```
int : keyword
 i : identifier
;
 for : keyword
(
 i : identifier
0: number
;
 i : identifier
=<= relational operator
24: number
;
 i : identifier
+
+
)
 if : keyword
(
 strcmp : keyword
(
 s : identifier
,
 keyword : identifier
[
 i : identifier
]
)
== relational operator
0: number
)
 return : keyword
1: number
;
 return : keyword
0: number
;
 my : identifier
 name : identifier
 is : identifier
 lehar : identifier
```

Console Output(Symbol table)



```
SRNO                    ID                      type
 0                      int                     keyword
 1                      i                       identifier
 2                      for                     keyword
 3                      i                       identifier
 4                      i                       identifier
 5                      i                       identifier
 6                      if                      keyword
 7                      strcmp                  keyword
 8                      s                       identifier
 9                      keyword                 identifier
 10                     i                       identifier
 11                     return                  keyword
 12                     return                  keyword
 13                     my                      identifier
 14                     name                    identifier
 15                     is                      identifier
 16                     lehar                   identifier


...Program finished with exit code 0
Press ENTER to exit console.
```