Ahsanullah University of Science and Technology (AUST)

Department of Computer Science and Engineering

Course No : CSE4130

Course Title: Formal Languages and Compilers Lab

Session : Spring 2020

Assignment No : 03

Submitted By :

Lab Group : A1

Name : Umme Habiba

ID :170104004

Question: Suppose, a given C source program has been scanned, filtered and then lexically analyzed 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. Now we generate a Symbol Table describing the features of the identifiers. Then, we generate a modified token stream in accordance with the Symbol Table for processing by the next phase, that is, Syntax Analysis.

Answer:

```
#include <stdio.h>

#include <string.h>

#include <stdbool.h>

char key_word[][20] = {"int", "char", "float", "double"};

int i;

FILE *p1,*p2;

int st_size = 0;

struct sTable{

   int id;

   char name[20];

   char idType[20];

   char dataType[20];

   char scope[20];

};

struct sTable st[50];

void keepId(){

   char now, lex[20];

   int i;

   p1 = fopen("input1.txt", "r");
```

```
p2 = fopen("output1.txt","w");
if(!p1)
   printf("\nFile can't be opened!");
else {
   bool isId = false;
   now = fgetc(p1);
   while(now != EOF{
      i=0;
      int count = 0;
      while(now!=' ' && now!='[' && now!=']' && now!=EOF) {
         lex[i] = now;
         ++i;
         now=fgetc(p1);
         count++;
      }
      lex[i]='\0';
      if(count == 0 &&  (now == '[' || now == ']' || now == ' ')){
         fputc(now,p2);
         now = fgetc(p1);
      }
      else{
         if(!strcmp(lex,"id")) {
            fputs(lex,p2);
            fputc(now,p2);
            now = fgetc(p1);
```

```
            i=0;
            while(now!=' ' && now!='[' && now!=']' && now!=EOF){
                lex[i] = now;
                ++i;
                now=fgetc(p1);
            }
            lex[i]='\0';
            fputs(lex,p2);
        }
        else{
            now = fgetc(p1);
            i=0;
            while(now!=' ' && now!='[' && now!=']' && now!=EOF){
                lex[i] = now;
                ++i;
                now=fgetc(p1);
            }
            lex[i]='\0';
            fputs(lex,p2);
        }
    }
  }
}
fclose(p1);
fclose(p2);
```

```
}
void generateST(){
   char now, lex[20];
   int i;
   p1 = fopen("output1.txt", "r");
   if(!p1)
      printf("\nFile can't be opened!");
   else{
      bool isGlobal = true, isVar = true, varcheck = false, idfound = false;
      char type[20], scop[20] = "global", idname[20];
      now = fgetc(p1);
      while(now != EOF){
         i=0;
         int count = 0;
         while(now!=' ' && now!='[' && now!=']' && now!=EOF)   {
            lex[i] = now;
            ++i;
            now=fgetc(p1);
            count++;
         }
         lex[i]='\0';
         if(count == 0 &&  (now == '[' || now == ']' || now == ' '))  {
            now = fgetc(p1);
         }
         else {
```

```
if(!strcmp(lex,"id")){
   now = fgetc(p1);
   i=0;
   while(now!=' ' && now!='[' && now!=']' && now!=EOF) {
      lex[i] = now;
      ++i;
      now=fgetc(p1);
   }
   lex[i]='\0';
   strcpy(idname,lex);
   varcheck = true;
   idfound = true;
}
else {
   if(istype(lex))
      strcpy(type,lex);
   now = fgetc(p1);
   if(lex[0] == '}'){
      isGlobal = true;
      strcpy(scop,"global");
   }
   if(varcheck && lex[0] != '(')  {
      isVar = true;
   }
   if(varcheck && lex[0] == '(' && isGlobal)  {
```

```
                isVar = false;
                isGlobal = false;
            }
            if(idfound)  {
                insertST(st_size,idname,isVar,type,scop);
                if(!isVar)
                   strcpy(scop,idname);
                idfound = false;
                varcheck = false;
            }
          }
        }
      }
   }
   fclose(p1);
}
int istype(char lex[20]){
   for(i=0; i<5; i++)
      if(!strcmp(lex,key_word[i]))
         return 1;
   return 0;
}
void insertST(int id, char idname[20], bool isVar, char type[20], char scop[20]){
   bool alreadyexist = false;
   for( i=0; i<st_size; i++)  {
```

```
        if(!strcmp(st[i].name,idname) && !strcmp(st[i].scope,scop))  {
            alreadyexist = true;
        }
        else if((!strcmp(st[i].name,idname) && !strcmp(st[i].scope,"global")))
            alreadyexist = true;
    }
    if(!alreadyexist){
        int i = st_size;
        st[i].id = i+1;
        strcpy(st[i].name,idname);
        if(isVar)
            strcpy(st[i].idType,"var");
        else
            strcpy(st[i].idType,"func");
        strcpy(st[i].dataType,type);
        strcpy(st[i].scope,scop);
        st_size++;
    }
}
int searchST(char idname[20]){
    for(i=0; i<st_size; i++){
        if(!strcmp(st[i].name,idname))  {
            return i+1;
        }
    }
```

```
    return -1;
}
void replaceIdname(){
    char now, lex[20] ;
    int i, id;
    p1 = fopen("output1.txt", "r");
    p2 = fopen("output3.txt","w");
    if(!p1)
        printf("\nFile can't be opened!");
    else{
        bool isId = false;
        now = fgetc(p1);
        while(now != EOF)  {
            i=0;
            int count = 0;
            while(now!=' ' && now!='[' && now!=']' && now!=EOF)  {
                lex[i] = now;
                ++i;
                now=fgetc(p1);
                count++;
            }
            lex[i]='\0';
            if(count == 0 &&  (now == '[' || now == ']' || now == ' ')){
                fputc(now,p2);
                now = fgetc(p1);
```

```
        }
        else  {
           if(!strcmp(lex,"id"))  {
              fputs(lex,p2);
              fputc(now,p2);
              now = fgetc(p1);
              i=0;
              while(now!=' ' && now!='[' && now!=']' && now!=EOF)  {
                 lex[i] = now;
                 ++i;
                 now=fgetc(p1);
              }
              lex[i]='\0';
              id = searchST(lex);
              itoa(id,lex,10);
              fputs(lex,p2);
           }
           else{
              fputs(lex,p2);
           }
        }
     }
  }
fclose(p1);
fclose(p2);
```

```
}
void displayST(){
   printf("-----------------------------------------------------------------------------------------\n");
   printf("SI No.\t\t|Name\t\t|Type\t\t|Data Type\t|Scope\n");

   for( i=0; i<st_size; i++) {
       printf("-----------------------------------------------------------------------------------------\n");

printf("%10d\t|%10s\t|%10s\t|%10s\t|%10s\n",st[i].id,st[i].name,st[i].idType,st[i].dataT
ype,st[i].scope);
   }
  printf("-----------------------------------------------------------------------------------------\n");
}
void updateST(int i,int col,char replaceWith[20]){
   i--;
   if(col == 2)
      strcpy(st[i].idType,replaceWith);
   else if(col == 3)
      strcpy(st[i].dataType,replaceWith);
   else if(col == 4)
      strcpy(st[i].scope,replaceWith);
}
void deleteST(int i){
   for(; i<st_size; i++) {
      strcpy(st[i-1].name,st[i].name);
```

```
        strcpy(st[i-1].idType,st[i].idType);
        strcpy(st[i-1].dataType,st[i].dataType);
        strcpy(st[i-1].scope,st[i].scope);
    }
    st_size = st_size - 1;
}
int main(void){
    char c,lex[20];
    int i,x,col;
    keepId();
    p2 = fopen("output1.txt","r");
    printf("Step-1:\n\n");
    while((c=fgetc(p2))!=EOF)
    printf("%c",c);
    fclose(p2);
    generateST();
    p2 = fopen("output2.txt","w");
    printf("Step-2:\n\n");
    printf("--------------------------------------------------------------------------------\n");
     printf("SI No.\t\t|Name\t\t|Type\t\t|Data Type\t|Scope\n");
    for( i=0; i<st_size; i++) {
        printf("--------------------------------------------------------------------------------\n");

printf("%10d\t|%10s\t|%10s\t|%10s\t|%s\n",st[i].id,st[i].name,st[i].idType,st[i].dataTyp
e,st[i].scope);
```

```
    itoa(st[i].id,lex,10);
    fputs(lex,p2);
    fputc(' ',p2);
    fputs(st[i].name,p2);
    fputc(' ',p2);
    fputs(st[i].idType,p2);
    fputc(' ',p2);
    fputs(st[i].dataType,p2);
    fputc(' ',p2);
    fputs(st[i].scope,p2);
    fputc(' ',p2);
    fputs("\n",p2);
}
 printf("--------------------------------------------------------------------------------\n");
fclose(p2);
replaceIdname();
p2 = fopen("output3.txt","r");
printf("Step-4:\n\n");
while((c=fgetc(p2))!=EOF)
printf("%c",c);
fclose(p2);
printf("Step-3:\n");
while (true) {
    printf("\n1.Insert\n2.Update\n3.Delete\n4.Search\n5.Display\n");
    printf("Select Operation: ");
```

```
int op;
scanf("%d", &op);
if(op==1){
   printf("\nInsert Function:\n");
    int id;
    char name1[100], type1[100], dtype1[100], scope1[100];
    printf("\nId: ");
    scanf("%d", &id);
    printf("Name: ");
    scanf("%s", &name1);
    printf("Type: ");
    scanf("%s",&type1);
    printf("Data Type: ");
    scanf("%s", &dtype1);
    printf("Scope: ");
    scanf("%s",&scope1);
    insertST(id,name1,type1,dtype1,scope1);
}
else if(op==2)  {
    printf("\nUpdate Function:\n");
    printf("Id:");
    scanf("%d",&x);
    printf("Column no:");
    scanf("%d",&col);
    printf("Value:");
```

```
        scanf("%s",&lex);
        updateST(x,3,lex);
    }
    else if(op==3){
        printf("\nDelete Function:\n");
        printf("Id: ");
        scanf("%d",&x);
        deleteST(x);
    }
    else if(op==4)  {
        printf("\nSearch Function:\n");
        printf("Id name: ");
        scanf("%s",&lex);
        if(searchST(lex)>0) printf("found.\n");
        else printf("not found.\n");
    }
    else if(op==5)  {
        printf("\nDisplay Function:\n");
        displayST();
    }
    else if(op==6) break;
    else printf("Invalid Option");
  }
  return 0;
}
```