Ahsanullah University of Science and Technology

Department of Computer Science and Engineering

CSE 4130

Formal languages and Compilers lab

Assignment No: 03

Submitted By:

Name: Anika Tanzim

ID: 16.02.04.072

Group: B1

Date of Submission: **24 August 2020**

Code:

```
#include <stdio.h>

int Keyword(char *str);

FILE *f1,*f2,*f3;
int j;

struct table
{
   char name[50], idType[50], dType[50], sc[50],val[50];
}tables[1000];


int Keyword(char *str) {
   int s=0;
   if(
   (!strcmp(str, "char")) || (!strcmp(str, "float")) || (!strcmp(str, "double")) ||
   (!strcmp(str, "int"))  || (!strcmp(str, "bool"))  )
   {
      s=1;
   }
   return s;
}


void insert(){

   f1 = fopen("OutputStep1.txt", "r");


   char scope[50];
   char arr[100];
   char c;
   int i;

   strcpy(scope, "global");
```

```
j=0;
while((c=fgetc(f1))!=EOF)
{
   if(c=='[')
   {
      i=0;
      while((c=fgetc(f1))!=']' && c!=' ')
      {
         arr[i++]=c;
      }
      arr[i]='\0';

      if(Keyword(arr)){

         strcpy(tables[j].dType, arr);   //data type found


         // last c was ]
         c=fgetc(f1);    //[
         c=fgetc(f1);    //i
         c=fgetc(f1);    //d
         int k=0;
         while((c=fgetc(f1))!=']')
         {
            tables[j].name[k++]=c;
         }
         tables[j].name[k]='\0'; //name found

         // last c was ]

         strcpy(tables[j].sc, scope); // scope found
         c=fgetc(f1);    //[

         if((c=fgetc(f1))== '('){
            strcpy(tables[j].idType, "func");//id type found
            strcpy(scope, tables[j].name);

         }
         else if(c=='='){
```

```
            c=fgetc(f1);    //]
            c=fgetc(f1);    //[
            k=0;
            while((c=fgetc(f1))!=']')
            {

            tables[j].val[k++]=c;// value found
            }

            strcpy(tables[j].idType, "var");
         }
         else{
            strcpy(tables[j].idType, "var");

         }

          j++;

      }
      else if(!strcmp("}", arr))
      {
         strcpy(scope, "global");
      }
      if( !Keyword(arr) && !strcmp("=", arr))
      {
         c=fgetc(f1);
         c =fgetc(f1);
         int k=0;
         if(isdigit(c)|| c=='.'){
            tables[j-1].val[k++]=c;
            while((c=fgetc(f1))!=']')
            {
               tables[j-1].val[k++]=c; //value found
            }
         }
      }
   }
}
```

```
    fclose(f1);



}


void delete(){
    int i=0,k=0;
    for(i=0; i<j; i++)
    {
        if(!strcmp(tables[i].sc,"global")){

            strcpy(tables[k].name,tables[i].name);
            strcpy(tables[k].idType,tables[i].idType);
            strcpy(tables[k].dType,tables[i].dType);
            strcpy(tables[k].sc,tables[i].sc);
            strcpy(tables[k].val,tables[i].val);
            k++;

        }
    }
    j=k; //updated table row number assigned


}




void display(){
    f2 = fopen("OutputStep2.txt", "w");
    printf("\nID\tName\tType\tData Type\tScope\tValue\n");
    printf("-------------------------------------------------------\n");
    fprintf(f2,"ID\tName\tType\tData Type\tScope\tValue\n");


    int i=0;
```

```
    for(i=0; i<j; i++)
    {

printf("%d\t%s\t%s\t%s\t\t%s\t%s\n",i+1,tables[i].name,tables[i].idType,tables[i].dType,tables[i]
.sc,tables[i].val);

fprintf(f2,"%d\t%s\t%s\t%s\t\t%s\t%s\n",i+1,tables[i].name,tables[i].idType,tables[i].dType,tabl
es[i].sc,tables[i].val);

    }


    fclose(f2);
    printf("\n\n");



}



int main(){

    char c;
    f1 = fopen("input.txt", "r");
    f2 = fopen("OutputStep1.txt", "w");



    //step 1
    char str[1000000];
    fscanf(f1,"%[^\n]",str);

    char *t= strtok(str, " ");

    printf(" result of step 1: \n\n");
    while(t != NULL){
       if(strcmp(t,"[kw") ==0|| strcmp(t,"[op")==0 ||strcmp(t,"[num")==0 || strcmp(t,"[sep")==0 ||
          strcmp(t,"[unkn")==0 || strcmp(t,"[sep")==0 || strcmp(t,"[par")==0 || strcmp(t,"[brc")==0){
```

```
            printf("[");
            fputc('[',f2);
        }
        else{
            printf("%s",t);
            fprintf(f2,"%s",t);

            if(strcmp(t,"[id") ==0){
                printf(" ");
                fputc(' ',f2);
            }

        }
        t= strtok(NULL, " ");


    }
    fclose(f1);
    fclose(f2);

    printf("\n\n");

    //step 2 & 3 --> symbol table

    printf("\nresult of step 2: \n\n");
    insert();
    display();
    printf("\nresult of step 3: (after deleting) \n\n");
    delete();
    display();


    return 0;


}
```

Output:

result of step 1:

[float][id x1][=][3.125][;][double][id f1][(][int][id x][)][{][double][id z][;][id z][=][0.01][;][return][id z][;][}][int][id main][(][void][)][{][int][id n1][;][double][id z][;][id n1][=][25][;][id z][=][id f1][(][id n1][)][;]

result of step 2:

ID	Name	Type	Data Type	Scope	Value
1	x1	var	float	global	3.125
2	f1	func	double	global	
3	x	var	int	f1	
4	z	var	double	f1	0.01
5	main	func	int	global	
6	n1	var	int	main	
7	z	var	double	main	25

result of step 3: (after deleting)

ID	Name	Type	Data Type	Scope	Value
1	x1	var	float	global	3.125
2	f1	func	double	global	
3	main	func	int	global	