Ahsanullah University of Science and Technology

Department of Computer Science and Engineering

CSE 4130

Formal languages and Compilers lab

Assignment No: 02

Submitted By:

Name: Anika Tanzim

ID: 16.02.04.072

Group: B1

Date of Submission: **8 March, 2020**

Question:

Suppose, we have a C source program scanned and filtered as it was done in Session 1. We now take that modified file as input, and separate the lexemes first. We further recognize and mark the lexemes as different types of tokens like keywords, identifiers, operators, separators, parenthesis, numbers, etc.

Answer:

```
#include <stdio.h>

#include <string.h>


void Lexemes();

int Keyword(char *str);

int Operator(char *str);

int Separator(char lexeme[]);

int Identifier(char lexeme[]);

int Parenthesis(char lexeme[]);

int Number(char lexeme[20]);


FILE *f1,*f2,*f3;


void Lexemes()

{

   char c;

   while((c=fgetc(f1))!=EOF)
```

```
{
   if(!isalnum(c) && c!='.' && c!=' ' && c!='_' ){
      fputc(' ', f2);
   }

   fputc(c, f2);
   if(c=='=' || c=='!' || c=='>' || c=='<' )
   {
      char d;
      if((d=fgetc(f1))=='=')
      {
         fputc(d, f2);
         fputc(' ', f2);
      }
      else
      {
         fputc(' ', f2);
         fputc(d, f2);
         if(!isalnum(d) && d!='.' && d!=' ' && d!='_')
            fputc(' ', f2);
      }
   }
```

```
        else if(!isalnum(c) && c!='.' && c!=' ' && c!='_')
            fputc(' ', f2);
    }

}

int Keyword(char *str) {
    int s=0;
    if( (!strcmp(str, "while")) || (!strcmp(str, "static")) || (!strcmp(str, "if")) ||
    (!strcmp(str, "volatile")) || (!strcmp(str, "do")) || (!strcmp(str, "goto")) ||
    (!strcmp(str, "sizeof")) || (!strcmp(str, "else")) || (!strcmp(str, "default")) ||
    (!strcmp(str, "void")) || (!strcmp(str, "for")) || (!strcmp(str, "signed")) ||
    (!strcmp(str, "continue")) || (!strcmp(str, "unsigned")) || (!strcmp(str, "short")) ||
    (!strcmp(str, "char")) || (!strcmp(str, "float")) || (!strcmp(str, "double")) ||
    (!strcmp(str, "int")) || (!strcmp(str, "char")) || (!strcmp(str, "const")) ||
    (!strcmp(str, "union")) || (!strcmp(str, "return")) || (!strcmp(str, "extern")) ||
    (!strcmp(str, "enum")) || (!strcmp(str, "register")) || (!strcmp(str, "typedef")) ||
    (!strcmp(str, "switch")) || (!strcmp(str, "long")) || (!strcmp(str, "break")) ||
    (!strcmp(str, "auto")) || (!strcmp(str, "struct")) )
    {
        s=1;
    }
```

```
    return s;
}



int Operator(char *str) {
    int s=0;
    if( (!strcmp(str, "+=")) || (!strcmp(str, "--")) || (!strcmp(str, "++")) ||
      (!strcmp(str, "<=")) || (!strcmp(str, "-=")) || (!strcmp(str, ">=")) ||
      (!strcmp(str, "&&")) || (!strcmp(str, "||")) || (!strcmp(str, "+")) ||
      (!strcmp(str, "-")) || (!strcmp(str, "=")) || (!strcmp(str, "==")) ||
      (!strcmp(str, "*")) || (!strcmp(str, "/")) || (!strcmp(str, "/=")) ||
      (!strcmp(str, "*=")) || (!strcmp(str, "%")) || (!strcmp(str, "%=")) ||
      (!strcmp(str, "!")) || (!strcmp(str, "^")) || (!strcmp(str, "<")) ||
      (!strcmp(str, ">"))){
          s=1;
      }
      return s;
}
int Separator(char *str)
{
    int i, l;
    l=strlen(str);
```

```
    if(l==1 && (str[0]==';' || str[0]=='"' || str[0]==',' || str[0]=='\'))
        return 1;
    else
        return 0;
}

int Parenthesis(char *str)
{
    int l = strlen(str);
    return (l==1 && (str[0]=='(' || str[0]==')' || str[0]=='{' || str[0]=='}' || str[0]=='[' || str[0]==']'));
}

int Identifier(char *str)
{
    int i, s=0, l;
    l=strlen(str);
    if((isalpha(str[0])) || (str[0]=='_'))
        s=1;
    if(s==1)
    {
        for(i=1; i<l; i++)
        {
```

```
        if(!isalnum(str[i]) && str[i]!='_'&& !isalpha(str[i]))
        {
            s=0;
            break;
        }
      }
   }
   return s;
}

int Number(char *str)
{
   int i, l, s;
   i=0;
   if(isdigit(str[i]))

   {
      s=1;
      i++;
   }
   else if(str[i]=='.')
   {
```

```
    s=2;
    i++;
}
else s=0;
l=strlen(str);
if(s==1)
    for(; i<l; i++)
    {
        if(isdigit(str[i]))
            s=1;
        else if(str[i]=='.')
        {
            s=2;
            i++;
            break;
        }
        else
        {
            s=0;
            break;
        }
    }
```

```
    if(s==2)
        if(isdigit(str[i]))
        {
            s=3;
            i++;
        }
        else s=0;
    if(s==3)
        for(; i<l; i++)
        {
            if(isdigit(str[i]))
                s=3;
            else
            {
                s=0;
                break;
            }
        }
    if(s==3) s=1;
    return s;
}
```

```
int main()
{
    char c;
    f1 = fopen("input.c", "r");
    f2 = fopen("lexemesOutput.txt", "w");

    Lexemes();

    fclose(f1);
    fclose(f2);

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

    while((c=fgetc(f1))!=EOF)
    {
       printf("%c",c);
    }
    printf("\n");
    f1 = fopen("lexemesOutput.txt", "r");
    f2 = fopen("tokensOutput.txt", "w");
    char str[50];
    while(fscanf(f1, "%s", &str)!=EOF)
```

```
{
    if(Keyword(str)){
        fprintf(f2, "[kw %s] ", str);}
    else if(Operator(str)){
        fprintf(f2, "[op %s] ", str);}
    else if(Separator(str)){
        fprintf(f2, "[sep %s] ", str);}
    else if(Identifier(str)){
        fprintf(f2, "[id %s] ", str);}
    else if(Parenthesis(str)){
        fprintf(f2, "[par %s] ", str);}
    else if(Number(str)){
        fprintf(f2, "[num %s]", str);}
    else
    {
        fprintf(f2, "[unkn %s]", str);
        f3 =fopen("ErrorOutput.txt","w");
        fprintf(f3, "\nunknown %s", str);
        fclose(f3);
    }
}
fclose(f1);
```

```
    fclose(f2);


    f1 = fopen("tokensOutput.txt", "r");
    while((c=fgetc(f1))!=EOF)
    {
        printf("%c",c);
    }
    printf("\n");


    f2 =fopen("Error.txt","r");
    while((c=fgetc(f2))!=EOF)
    {
        printf("%c",c);
    }
    fclose(f2);
    return 0;
}
```

Input:

char c; int x1,x_2; float y1,y2;x1=5;x_2=
10;y1=2.5+x1*45;y2=100.o5-x_2/3;if(y1<=y2)c='y';else c='n';

Output:

char c ; int x1 , x_2 ; float y1 , y2 ; x1 = 5 ; x_2 = 10 ; y1 = 2.5 + x1 * 45 ; y2 = 100.o5 - x_2 / 3 ; if (y1 <= y2) c = ' y ' ; else c = ' n ' ;

[kw char] [id c] [sep ;] [kw int] [id x1] [sep ,] [id x_2] [sep ;] [kw float] [id y1] [sep ,] [id y2] [sep ;] [id x1] [op =] [num 5][sep ;] [id x_2] [op =] [num 10][sep ;] [id y1] [op =] [num 2.5][op +] [id x1] [op *] [num 45][sep ;] [id y2] [op =] [unkn 100.o5][op -] [id x_2] [op /] [num 3][sep ;] [kw if] [par (] [id y1] [op <=] [id y2] [par)] [id c] [op =] [sep '] [id y] [sep '] [sep ;] [kw else] [id c] [op =] [sep '] [id n] [sep '] [sep ;]

Error : unknown 100.o5