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 : 02

Submitted By :

Lab Group : A1

Name : Umme Habiba

ID :170104004

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>

FILE *p1,*p2,*p3;

int checkSeparator(char lex[])

{

   int i, l;

   l=strlen(lex);

   if(l==1 && (lex[0]==',' || lex[0]==39 || lex[0]==';' || lex[0]=='"'))

      return 1;

   else

      return 0;

}

int checkOperator(char lex[])

{

   int i, l,s=0;

   l=strlen(lex);

   if(l==1 && (lex[0]=='>' || lex[0]=='+' || lex[0]=='-' || lex[0]=='*' || lex[0]=='/'
|| lex[0]=='<' || lex[0]=='='))
```

```
    return 1;
  else if(l==2 && (lex[0]=='>' || lex[0]=='<'|| lex[0]=='=' || lex[0]=='!') &&
lex[1]=='=')
    s=1;

  return s;
}
int checkParenthesis(char lex[])
{
  int l = strlen(lex);
  return (l==1 && (lex[0]=='(' || lex[0]==')' || lex[0]=='{' || lex[0]=='}' ||
lex[0]=='[' || lex[0]==']'));
}
int checkNumber(char lex[])
{
  int i, l, s;
  i=0;
  if(isdigit(lex[i]))
 {
    s=1;
    i++;
  }
  else if(lex[i]=='.')
```

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

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

```
    s=1;
  return s;
}
int checkId(char lex[])
{
  int i, s=0, l;
  l=strlen(lex);
  if((isalpha(lex[0])) || (lex[0]=='_'))
    s=1;
  if(s==1)
  {
    for(i=1; i<l; i++)
    {
      if(!isalnum(lex[i]) && lex[i]!='_'&& !isalpha(lex[i]))
      {
        s=0;
        break;
      }
    }
  }
  return s;
}
```

```
int keyCheck(char lex[])
{
   int i, s=0;
   char valid_key[][15] = {"float","int","double","char","return","if","else","else if","while","for","case","do while","goto","break","continue","switch"};
   for(i=0; i<15; i++)
   {
      if(!strcmp(lex,valid_key[i]))
      {
         s=1;
         break;
      }
   }
   return s;
}

int main()
{
   char c;
   p1 = fopen("input1.txt", "r");
   p2 = fopen("lexemes.txt", "w");
   //separation
   while((c=fgetc(p1))!=EOF)
```

```
{
   if(!isalnum(c) && c!=' ' && c!='_' && c!='.')
      fputc(' ', p2);
   fputc(c, p2);
   if(c=='>' || c=='<'|| c=='=' || c=='!')
   {
       char ch=fgetc(p1);

      if(ch=='=')
      {
         fputc(ch, p2);
         fputc(' ', p2);
      }
      else
      {
         fputc(' ', p2);
         fputc(ch, p2);
         if(!isalnum(ch) && ch!=' ' && ch!='_' && ch!='.')
            fputc(' ', p2);
      }
   }
   else if(!isalnum(c) && c!=' ' && c!='_' && c!='.')
```

```
        fputc(' ', p2);
}

fclose(p1);
fclose(p2);

printf("\nSample Input :\n\n");
p1 = fopen("input1.txt", "r");
while((c=fgetc(p1))!=EOF)
{
   printf("%c",c);
}
printf("\n\n");
p1 = fopen("lexemes.txt", "r");
printf("\nStep 1 : \n\n");
while((c=fgetc(p1))!=EOF)
{
   printf("%c",c);
}
printf("\n\n\n");
p1 = fopen("lexemes.txt", "r");
p2 = fopen("tokens.txt", "w");
```

```
char str[1000];
while(fscanf(p1, "%s", &str)!=EOF)
{
   if( keyCheck(str))
      fprintf(p2, "[kw %s] ", str);
   else if(checkOperator(str))
      fprintf(p2, "[op %s] ", str);
   else if(checkParenthesis(str))
      fprintf(p2, "[par %s] ", str);
   else if(checkSeparator(str))
      fprintf(p2, "[sep %s] ", str);
   else if(checkId(str))
      fprintf(p2, "[id %s] ", str);
   else if(checkNumber(str))
      fprintf(p2, "[num %s]", str);
   else
   {
      fprintf(p2, "[unkn %s]", str);
      p3 =fopen("Error.txt","w");
      fprintf(p3, "\nError Part : unknown %s", str);
      fclose(p3);
   }
```

```
    }
    fclose(p1);
    fclose(p2);
    printf("Step 2 : \n\n");
    p1 = fopen("tokens.txt", "r");
    while((c=fgetc(p1))!=EOF)
    {
       printf("%c",c);
    }
    printf("\n\n");
    p3 =fopen("Error.txt","r");
    while((c=fgetc(p3))!=EOF)
    {
       printf("%c",c);
    }
    fclose(p3);
    printf("\n\n");
    return 0;
}
```