

**AHSANULLAH UNIVERSITY OF**

**SCIENCE AND TECHNOLOGY**

**Course Name:** Formal Languages & Compilers Lab

**Course No:** CSE 4130

**Assignment No:** 02

**Name:** Ahmad Subaktagin Jabir

**ID:** 16.02.04.061

**Section:** B

**Group:** B1

**Date of Submission:** March 08, 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.

**Ans:**

**Code:**

#include<stdio.h>

int isKeyword(char buffer[])

{

char keywords[32][10]= {"auto","break","case","char","const","continue","default",

"do","double","else","enum","extern","float","for","goto",

"if","int","long","register","return","short","signed",

"sizeof","static","struct","switch","typedef","union",

"unsigned","void","volatile","while"

};

int flag=0;

for (int i=0; i<32; i++)

{

if(strcmp(keywords[i],buffer)==0)

{

flag=1;

break;

}

}

return flag;

}

int isInteger(char\* ch)

{

if(strlen(ch)==0)

return 0;

for(int i=0; i<strlen(ch); i++)

{

if(ch[i]!='0' && ch[i]!='1' && ch[i]!='2'

&& ch[i]!='3' && ch[i]!='4' && ch[i]!='5'

&& ch[i]!='6' && ch[i]!='7' && ch[i]!='8'

&& ch[i]!='9' || (ch[i]!='-' && i>0))

return 0;

}

return 1;

}

int isFloat(char buffer[])

{

int flag=0;

for(int i=0; i<strlen(buffer); i++)

{

if(buffer[i]!='0' && buffer[i]!='1' && buffer[i]!='2'

&& buffer[i]!='3' && buffer[i]!='4' && buffer[i]!='5'

&& buffer[i]!='6' && buffer[i]!='7' && buffer[i]!='8'

&& buffer[i]!='9' && buffer[i]!='.' || (buffer[i]!='-' && i>0))

return 0;

if(buffer[i]=='.')

flag=1;

}

return flag;

}

int isIdentifier(char\* ch)

{

if(ch[0]=='0'||ch[0]=='1'||ch[0]=='2'

||ch[0]=='3'||ch[0]=='4'||ch[0]=='5'

||ch[0]=='6'||ch[0]=='7'||ch[0]=='8'

||ch[0]=='9')

return 0;

return 1;

}

int isOperator(char ch)

{

if(ch=='+'||ch=='-'||ch=='\*'

||ch=='/'||ch=='%'||ch=='='

||ch=='<'||ch=='>')

return 1;

return 0;

}

int isParenthesis(char ch)

{

if(ch=='{'||ch=='('||ch=='['||

ch=='}'||ch==')'||ch==']')

return 1;

return 0;

}

int main()

{

char ch, buffer[20], quotation[]="''";

FILE \*p1, \*p2;

int i, j=0, k=0;

p1= fopen("input.c","r");

p2= fopen("output.txt","w");

if(!p1)

printf("\nFile can't be opened!");

else

{

while((ch=fgetc(p1))!=EOF)

{

if(isalnum(ch))

buffer[j++]=ch;

else if(ch=='\_'||ch=='.')

buffer[j++]=ch;

if((ch==' '||ch==','||ch=='('||ch==')'

||ch=='='||ch=='+'||ch=='-'||ch=='\*'

||ch=='<'||ch=='>'||ch=='\n'||ch==';'

||ch==quotation[0]||ch=='/') && (j!=0))

{

buffer[j]='\0';

j=0;

if(isKeyword(buffer)==1)

{

printf("[kw %s] ", buffer);

fprintf(p2,"[kw ");

for(i=0; i<strlen(buffer); i++)

{

fprintf(p2,"%s", buffer);

break;

}

fprintf(p2,"] ");

}

else

{

if(isFloat(buffer)==1)

{

printf("[num %s] ", buffer);

fprintf(p2,"[num ");

for(i=0; i<strlen(buffer); i++)

{

fprintf(p2,"%s", buffer);

break;

}

fprintf(p2,"] ");

}

else if(isInteger(buffer)==1)

{

printf("[num %s] ", buffer);

fprintf(p2,"[num ");

for(i=0; i<strlen(buffer); i++)

{

fprintf(p2,"%s", buffer);

break;

}

fprintf(p2,"] ");

}

else if(isIdentifier(buffer)==1)

{

printf("[id %s] ", buffer);

fprintf(p2,"[id ");

for(i=0; i<strlen(buffer); i++)

{

fprintf(p2,"%s",buffer);

break;

}

fprintf(p2,"] ");

}

else

{

printf("[unkn %s] ", buffer);

fprintf(p2,"[unkn ");

for(i=0; i<strlen(buffer); i++)

{

fprintf(p2,"%s", buffer);

break;

}

fprintf(p2,"] ");

}

}

}

for(k=0; k<6; ++k)

{

if(isParenthesis(ch)==1)

{

printf("[par %c] ",ch);

fprintf(p2,"[par %c] ",ch);

break;

}

}

for(i=0; i<8; i++)

{

if(isOperator(ch)==1)

{

printf("[op %c] ",ch);

fprintf(p2,"[op %c] ",ch);

break;

}

}

if(ch==';'||ch==','||ch==quotation[0])

{

printf("[sep %c] ",ch);

fprintf(p2,"[sep %c] ",ch);

}

}

fclose(p1);

fclose(p2);

}

return 0;

}

**Input:**

#include<stdio.h> int main(void) { printf ("Hello"); printf("World"); return 0; }

**Output:**

[id include] [op <] [id stdio.h] [op >] [kw int] [id main] [par (] [kw void] [par )] [par {] [id printf] [par (] [id Hello] [par )] [sep ;] [id printf] [par (] [id World] [par )] [sep ;] [kw return] [num 0] [sep ;] [par }]