%{

#include<stdio.h>

#include<string.h>

FILE \*fp,\*yyin;

int i,symcnt=0;

struct Symboltable

{

int index;

char name[20];

}s[10];

%}

%%

"#include"[\t]\*"<"[a-z]+"."[hc]">" {printf("\nPreprocessor directive %s",yytext);}

"int"|"main"|"printf"|"scanf"|"if"|"return"|"else"|"for" {printf("\nKeyword %s",yytext);}

[{}()]+ {printf("\nParenthesis %s",yytext);}

[,;:] {printf("\nDelimiter %s",yytext);}

[-+\*/=&] {printf("\nOperator %s",yytext);}

[0-9]+ {printf("\nConstant %s",yytext);}

\"[^\n"]\*\" {printf("\nString constant %s",yytext);}

[a-zA-Z][a-zA-Z0-9]\* {printf("\nIdentifier %s",yytext); insert(yytext);}

%%

int insert(char txt[20])

{

int flag=0;

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

{

if((strcmp(s[i].name,txt))==0)

{

flag=1;

break;

}

}

if(flag!=1)

{

symcnt++;

strcpy(s[symcnt].name,txt);

s[symcnt].index=symcnt;

}

return 0;

}

int main()

{

fp=fopen("add.c","r");

int j;

yyin=fp;

yylex();

printf("\nIndex:");

printf("\tName:\n");

for(j=1;j<=symcnt;j++)

{

printf("\n%d",s[j].index);

printf("\t%s\n",s[j].name);

}

return 0;

}

int yywrap()

{

return 1;

}

**INPUT FILE:**

#include<stdio.h>

int main()

{

int a , b , c ;

printf("\n Enter the first number : ") ;

scanf("%d" , & a );

printf("\n Enter the second number : ") ;

scanf("%d" , & b ) ;

c = a + b ;

printf("\nAddition of two numbers:%d\n" , c) ;

return 0 ;

}

**OUTPUT:**

pccoe@212A-16:~/Ashwini$ lex rule.l

pccoe@212A-16:~/Ashwini$ gcc lex.yy.c

pccoe@212A-16:~/Ashwini$ ./a.out

Preprocessor directive #include<stdio.h>

Keyword int

Keyword main

Parenthesis ()

Parenthesis {

Keyword int

Identifier a

Delimiter ,

Identifier b

Delimiter ,

Identifier c

Delimiter ;

Keyword printf

Parenthesis (

String constant "\n Enter the first number : "

Parenthesis )

Delimiter ;

Keyword scanf

Parenthesis (

String constant "%d"

Delimiter ,

Operator &

Identifier a

Parenthesis )

Delimiter ;

Keyword printf

Parenthesis (

String constant "\n Enter the second number : "

Parenthesis )

Delimiter ;

Keyword scanf

Parenthesis (

String constant "%d"

Delimiter ,

Operator &

Identifier b

Parenthesis )

Delimiter ;

Identifier c

Operator =

Identifier a

Operator +

Identifier b

Delimiter ;

Keyword printf

Parenthesis (

String constant "\nAddition of two numbers:%d\n"

Delimiter ,

Identifier c

Parenthesis )

Delimiter ;

Keyword return

Constant 0

Delimiter ;

Parenthesis }

Index: Name:

pccoe@212A-16:~/Ashwini$ gcc lex.yy.c

pccoe@212A-16:~/Ashwini$ lex rule.l

pccoe@212A-16:~/Ashwini$ gcc lex.yy.c

pccoe@212A-16:~/Ashwini$ ./a.out

Preprocessor directive #include<stdio.h>

Keyword int

Keyword main

Parenthesis ()

Parenthesis {

Keyword int

Identifier a

Delimiter ,

Identifier b

Delimiter ,

Identifier c

Delimiter ;

Keyword printf

Parenthesis (

String constant "\n Enter the first number : "

Parenthesis )

Delimiter ;

Keyword scanf

Parenthesis (

String constant "%d"

Delimiter ,

Operator &

Identifier a

Parenthesis )

Delimiter ;

Keyword printf

Parenthesis (

String constant "\n Enter the second number : "

Parenthesis )

Delimiter ;

Keyword scanf

Parenthesis (

String constant "%d"

Delimiter ,

Operator &

Identifier b

Parenthesis )

Delimiter ;

Identifier c

Operator =

Identifier a

Operator +

Identifier b

Delimiter ;

Keyword printf

Parenthesis (

String constant "\nAddition of two numbers:%d\n"

Delimiter ,

Identifier c

Parenthesis )

Delimiter ;

Keyword return

Constant 0

Delimiter ;

Parenthesis }

Index: Name:

1 a

2 b

3 c

pccoe@212A-16:~/Ashwini$