CD Assignment 3

--------------------------------

Name: Yash Oswal

Div: B Roll no.: 38

SRN: 201901226

--------------------------------

Input:

kwd void main int float printf

int a = 5; a = a + 2; printf(a);

Output:



Source Code:

%{

enum {

LOOKUP =0, /\* default - looking rather than defining. \*/

KWD,

IDE,

OPT,

DEL

};

int state;

int add\_word(int type, char \*word);

int lookup\_word(char \*word);

%}

%%

\n { state = LOOKUP; }

^kwd { state = KWD; }

^ide { state = IDE; }

^opt { state = OPT; }

^del { state = DEL; }

[0-9] { printf("%s: Constant\n", yytext); }

[+-/\*=%] {

if(state != LOOKUP) {

add\_word(state, yytext);

} else {

switch(lookup\_word(yytext)) {

case OPT:

printf("%s: Oerator\n", yytext);

break;

default:

printf("%s: Identifier\n", yytext);

break;

}

}

}

[;'")(:] {

if(state != LOOKUP) {

add\_word(state, yytext);

} else {

switch(lookup\_word(yytext)) {

case DEL:

printf("%s: Delimiter\n", yytext);

break;

default:

printf("%s: Identifier\n", yytext);

break;

}

}

}

[a-zA-Z]+ {

if(state != LOOKUP) {

add\_word(state, yytext);

} else {

switch(lookup\_word(yytext)) {

case KWD:

printf("%s: Keyword\n", yytext);

break;

case IDE:

printf("%s: Identifier\n", yytext);

break;

default:

printf("%s: Identifier\n", yytext);

break;

}

}

}

. /\* ignore anything else \*/ ;

%%

int main()

{

yylex();

}

struct word {

char \*word\_name;

int word\_type;

struct word \*next;

};

struct word \*word\_list;

extern void \*malloc() ;

int add\_word(int type, char \*word)

{

struct word \*wp;

if(lookup\_word(word) != LOOKUP)

{

printf("!!! warning: word %s already defined \n", word);

return 0;

}

wp = (struct word \*) malloc(sizeof(struct word));

wp->next = word\_list;

wp->word\_name = (char \*) malloc(strlen(word)+1);

strcpy(wp->word\_name, word);

wp->word\_type = type;

word\_list = wp;

return 1;

}

int lookup\_word(char \*word)

{

struct word \*wp = word\_list;

for(; wp; wp = wp->next) {

if(strcmp(wp->word\_name, word) == 0)

return wp->word\_type;

}

return LOOKUP;

}