**ASSIGNMENT NO.**

NAME : SHRUTI DILIP BHUJANGE

CLASS : BE COMP 1 ROLL NO.:402006

PROGRAM:

%{

#include<string.h>

#include<stdio.h>

struct node

{

char name[20];

char type[20];

struct node \*next;

};

%}

sep ,

ter ;

dt "int"|"float"|"double"|"char"

L [a-zA-Z]

D [0-9]

id {L}({L}|{D})\*

con "if"|"else"|"while"|"for"

op "+"|"-"|"\*"|"/"|"="

logop "&&"|"||"|"!"

inc "++"

dec "--"

rel "=="|"!="|">"|"<"|"<="|">="

bit "&"|"|"|"~"

key "void"|"break"|"case"|"return"|"goto"

builtin "main"|"printf"|"scanf"

spc " "

cm [/][/]([a-zA-Z]|[ ]|[0-9])\*

cm2 [/][\*]([a-zA-Z]|[ ]|[0-9]|['\n'])\*[\*][/]

%%

{D}+ {printf("%s is a Number.\n",yytext);}

{dt} {printf("%s is a Datatype.\n",yytext);}

{con} {printf("%s is a Construct.\n",yytext);}

{key} {printf("%s is a Keyword.\n",yytext);}

{spc}

{id}("["{D}\*"]")+ {printf("%s is an Array.\n",yytext);insert("Array");}

{builtin}"()" {printf("%s is a Built-in Function.\n",yytext);}

{id}"()" {printf("%s is a User-defined Function.\n",yytext);insert("Function");}

{sep} {printf("%s is a Separator.\n",yytext);}

{ter} {printf("%s is a Terminator.\n",yytext);}

"){" {printf("%s is a starting of Function definition.\n",yytext);}

{op} {printf("%s is an Arithmetic Operator.\n",yytext);}

{logop} {printf("%s is a Logical Operator.\n",yytext);}

{rel} {printf("%s is a Relational Operator.\n",yytext);}

{bit} {printf("%s is a Bitwise Operator.\n",yytext);}

{inc} {printf("%s is an Increment Operator.\n",yytext);}

{dec} {printf("%s is an Decrement Operator.\n",yytext);}

{id} {printf("%s is an Identifier.\n",yytext);insert("Variable");}

{cm} {printf("%s is a Comment.\n",yytext);}

{cm2} {printf("%s is a Comment.\n",yytext);}

%%

struct node \*head;

int insert(char s[25])

{

struct node \*current = head;

int flag=0;

while(current->next!=NULL)

{

current=current->next;

if(strcmp(current->name,yytext)==0)

{

flag=1;

}

}

if(flag==0)

{

current->next=malloc(sizeof(struct node));

current=current->next;

strcpy(current->name,yytext);

strcpy(current->type,s);

current->next=NULL;

}

return 0;

}

void disp()

{

struct node \*ptr = head;

while(ptr!= NULL)

{

printf("%s",ptr->name);

printf("\t");

printf("%s",ptr->type);

printf("\n");

ptr = ptr->next;

}

}

int main()

{

head=(struct node\*)malloc(sizeof(struct node));

strcpy(head->name,"Symbol");

strcpy(head->type,"Type");

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

yylex();

yywrap();

disp();

fclose(yyin);

return 0;

}

int yywrap()

{

return 1;

}