**EXPERIMENT NO. 4**

**AIM:** Design and implementation of lexical analyser

**PROGRAM:**

#include<stdio.h>

#include<string.h>

#include<regex.h>

typedef struct{

char keyword[10];

}key;

typedef struct{

char operator[3];

}op;

key keywords[4];

op operators[2];

const char\* re = "[a-z][a-z1-9\_]\*[a-z1-9]\*";

const char\* const\_re = "[0-9]+\\.?[0-9]+";

FILE \*ptr;

int findKeyword(char \*str){

int i;

for(i=0;i<4;i++){

if(strcmp(keywords[i].keyword,str) == 0){

return 1;

}

}

return 0;

}

int findOperator(char \*str){

int i;

for(i=0;i<2;i++){

if(strcmp(operators[i].operator,str) == 0){

return 1;

}

}

return 0;

}

int match(char \*str, const char \*pattern){

regex\_t re;

if (regcomp(&re, pattern, REG\_EXTENDED|REG\_NOSUB) != 0) return 0;

int status = regexec(&re, str, 0, NULL, 0);

regfree(&re);

if (status != 0) return 0;

return 1;

}

int checkVar(char \*str){

if(!(str[0] >= 'a' && str[0] <= 'z')){

return 0;

}

if(match(str,re)){

return 1;

}

return 0;

}

int checkConst(char \*str){

if(match(str,const\_re)){

return 1;

}

return 0;

}

void main(){

char buffer[10];

int val;

strcpy(keywords[0].keyword,"int");

strcpy(keywords[1].keyword,"short");

strcpy(keywords[2].keyword,"print");

strcpy(keywords[3].keyword,"float");

strcpy(operators[0].operator,"=");

strcpy(operators[1].operator,"+");

ptr = fopen("lex.txt","r");

while(fscanf(ptr,"%s",buffer) == 1){

if(findKeyword(buffer)){

printf("%s keyword\n",buffer);

}

else if(findOperator(buffer)){

printf("%s operator\n",buffer);

}

else{

val = checkVar(buffer);

if(val){

printf("%s identifier\n",buffer);

}

else{

val = checkConst(buffer);

if(val){

printf("%s constant\n",buffer);

}

else{

printf("\*\*\*%s is an invalid token\*\*\*\n",buffer);

}

}

}

}

}

**OUTPUT:**

int keyword

a identifier

= operator

10 constant

short keyword

b identifier

= operator

\*\*\*5 is an invalid token\*\*\*

int keyword

c identifier

= operator

a identifier

+ operator

b identifier

print keyword

c identifier

float keyword

\*\*\*2a is an invalid token\*\*\*

= operator

5.0 constant