Compiler Design Lab WEEK 1

Experiment:

Implementation of lexical analyzer

Aim:

To implement lexical analyzer and check for tokens in the given program.

Algorithm:

Step 1: Declare the necessary variables.

Step2: Declare an array and store the keywords in that array

Step 3: Open the input file in read open

Step 4: read the string from the file till the end of file.

Step 4.1: If the first character in the strings # then print that string as header file.

Step 4.2: If the string matches with any of the keywords print. That string is a keyword

Step 4.3: If the string matches with operator and special Symbols print the corresponding message

Step 4.4: If the string is not a keyword then prints that as an identifier.

Program:

lex.c

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<ctype.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 i, flag = 0;

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

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

flag = 1;

break;

}

}

return flag;

}

int main(){

char ch, buffer[15], operators[] = "+-\*/%=";

FILE \*fp;

int i,j=0;

fp = fopen("program.txt","r");

if(fp == NULL){

printf("error while opening the file\n");

exit(0);

}

while((ch = fgetc(fp)) != EOF){

for(i = 0; i < 6; ++i){

if(ch == operators[i])

printf("%c is operator\n", ch);

}

if(isalnum(ch)){

buffer[j++] = ch;

}

else if((ch == ' ' || ch == '\n') && (j != 0)){

buffer[j] = '\0';

j = 0;

if(isKeyword(buffer) == 1)

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

else

printf("%s is indentifier\n", buffer);

}

}

fclose(fp);

return 0;

}

INPUT

program.txt

#include <stdio.h>

int main() {

int number1, number2, sum;

printf("Enter two integers: ");

scanf("%d %d", &number1, &number2);

// calculating sum

sum = number1 + number2;

printf("%d + %d = %d", number1, number2, sum);

return 0;

}

Output

