

DEVELOP A LEXICAL ANALYZER TO RECOGNIZE FEW PATTERNS IN C (EX. IDENTIFIERS, CONSTANTS, COMMENTS, OPERATOR ETC.) AND IMPLEMENTATION OF A SYMBOL TABLE

Ex.No:1

Date:

AIM:

To develop a lexical analyzer to recognize few patterns in C (Ex. Identifiers, Constants, Comments, Operators etc.) and Implementation of a symbol table.

ALGORITHM:

Step 1: Start the program

Step 2: Read the input string.

Step 3: Check whether the string is identifier, operator, symbol by using the rules of identifier and keywords using lex tool using the following steps.

Step 4: If the string starts with letter followed by any number of letter or digit then display it as a identifier.

Step 5: If it is operator print it as a operator

Step 6: If it is number print it as a number

Step 7: Stop the program

PROGRAM:

```
%{
#include <stdio.h>
%}

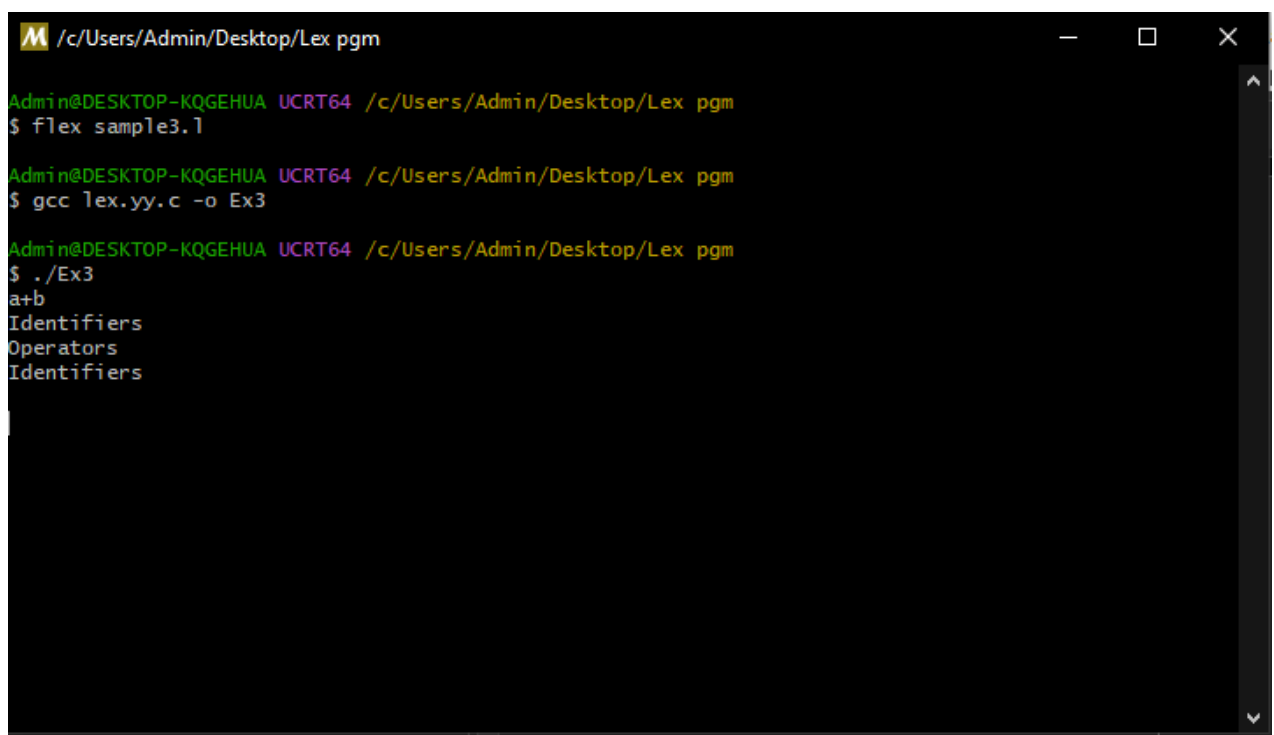
%%

bool|int|float|char printf("Keyword\n");
[-+]+      printf("Operators\n");
[0-9]+      printf("Numbers\n");
[.,'"]+     printf("Punctuation Chars\n");
[&%*$@!]+  printf("Special Characters\n");
[a-zA-Z]+   printf("Identifiers\n");
%%
```

```
int main() {  
    yylex();  
    return 0;  
}
```

```
int yywrap() {  
    return 1;  
}
```

OUTPUT:



```
M /c/Users/Admin/Desktop/Lex pgm  
Admin@DESKTOP-KQGEHUA UCRT64 /c/Users/Admin/Desktop/Lex pgm  
$ flex sample3.1  
Admin@DESKTOP-KQGEHUA UCRT64 /c/Users/Admin/Desktop/Lex pgm  
$ gcc lex.yy.c -o Ex3  
Admin@DESKTOP-KQGEHUA UCRT64 /c/Users/Admin/Desktop/Lex pgm  
$ ./Ex3  
a+b  
Identifiers  
Operators  
Identifiers
```

CONCLUSION:

Thus to develop a lexical analyzer to recognize few patterns in C (Ex. Identifiers, Constants, Comments, Operators etc.) was executed successfully