GOA COLLEGE OF ENGINEERING

"Bhausaheb Bandodkar Technical Education Complex"

Experiment No: 4

Lexical Analyzer-I

Aim: Write a Lex program to validate following tokens such as Identifier, Floating Point Constant, Email Id for given input and read the input from text file.

Theory:

The first phase in a compiler reads the input source and converts strings in the source to tokens. Using regular expressions, we can specify patterns to lex that allow it to scan and match strings in the input. Each pattern in lex has an associated action. Typically an action returns a token, representing the matched string, for subsequent use by the parser. To begin with, however, we will simply print the matched string rather than return a token value. We may scan for identifiers using the regular expression

Lex Program:

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

%%
if|else|while|int|switch|for|char {printf("Valid Keyword\n");}
[-+]?[0-9]* {printf("Valid Integer\n");}
[-+]?[0-9]*[.][0-9]+ {printf("Valid Float\n");}
[a-zA-Z_][a-zA-Z0-9_]* {printf("Valid Identifier\n");}
[0-9_][a-zA-Z0-9_]* {printf("Invalid Identifier\n");}
[a-z0-9]+@[a-z]+\.[a-z]{2,3} {printf("Valid Email\n");}
[\t\n]+ ;
%%

int main(){
   yyin=fopen("sample.txt","r");
   yylex();
}

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

Conclusion:

The lex program to validate tokens such as identifiers, numeric number, etc. from a file has been successfully executed.