**Aim:** Write a LEX program to convert substring abc to ABC from the given input string

**Theory:** Lex is a computer program that generates lexical analyzers ("scanners" or "lexers"). Lex is commonly used with the yacc parser generator. Lex, originally written by Mike Lesk and Eric Schmidt and described in 1975, is the standard lexical analyzer generator on many Unix systems, and an equivalent tool is specified as part of the POSIX standard.

Lex reads an input stream specifying the lexical analyzer and writes source code which implements the lexical analyzer in the C programming language.

In addition to C, some old versions of Lex could also generate a lexer in Ratfor.

Some major points to keep in mind about LEX is:

* Lex is a program that generates lexical analyzer. It is used with YACC parser generator.
* The lexical analyzer is a program that transforms an input stream into a sequence of tokens.
* It reads the input stream and produces the source code as output through implementing the lexical analyzer in the C program.

**Code:**

%{

#include<stdio.h>

#include<string.h>

int i;

%}

%%

[a-z A-Z]\* {

for(i=0;i<=yyleng;i++)

{

if((yytext[i]=='a')&&(yytext[i+1]=='b')&&(yytext[i+2]=='c'))

{

yytext[i]='A';

yytext[i+1]='B';

yytext[i+2]='C';

}

}

printf("%s",yytext);

}

[\t]\* return;

.\* {ECHO;}

\n {printf("%s",yytext);}

%%

main()

{

yylex();

}

int yywrap()

{

return 1;

}

**Output:**

[CSE@localhost ~]$ lex lex1.l

[CSE@localhost ~]$ cc lex.yy.c

[CSE@localhost ~]$. /a.out

 abc

ABC