Exp 4 : SUBSTRING abc TO ABC

1. Start
2. Lex Section Rules
    1. For each occurrence of [a-zA-Z]*abc replace each 'a' with 'A', 'b' with 'B', and 'c' with 'C'
3. yywrap Function - Return 1 to indicate end of input

4. Main Program
    1. Print a prompt to enter input
    2. Call yylex to initiate lexical analysis
5. Stop

substring ABC code (Lex)

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

%%
[a-zA-Z]*abc {
   for(int 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);
}
%%

int yywrap(){
   return 1;
}

int main(){
   printf("Enter Input : ");
   yylex();
   return 0;
}
```

output

```
deadpool@daredevil:~/Desktop/s7-CD/02 LEX/Substring ( abc )$ flex abc_to_ABC.l
deadpool@daredevil:~/Desktop/s7-CD/02 LEX/Substring ( abc )$ gcc lex.yy.c -o abc
deadpool@daredevil:~/Desktop/s7-CD/02 LEX/Substring ( abc )$ ./abc
Enter Input : this program converts abc to ABC ie substringabc
this program converts ABC to ABC ie substringABC
deadpool@daredevil:~/Desktop/s7-CD/02 LEX/Substring ( abc )$
```