Principle of Compiler Construction

(COCSC14)

Viva File

Submitted by:
Ashish Kumar
2019UCO1518

To:
Ms. Rahila Khan

Ques 1. Construct a grammar for Roman Numerals

Grammar :

```
Context Free Grammar for Roman Numerals can be given as G = (V, T, S,
P),  where we have:

V = Set of non-terminals
   = { numeral, digit(decimal_place), digit(n), n, 0(n), 1(n), 2(n), 3(n),
4(n), 5(n), 6(n), 7(n), 8(n),9(n), x(n), s(n), x(1),
    x(2), x(3), x(4), s(1), s(2), s(3) }

T = Set of terminals
   = {characters that combine to form roman numerals}
   = { I, X, C, M, V, L, D }

S = Start Symbol
   = roman_numeral

P = Set of productions :-

roman_numeral    ->     digits(decimal_place)
decimal_place     ->    n
digit(n)    ->    0(n) | 1(n) | 2(n) | 3(n) | ... | 9(n)

0(n)     ->     <empty>
1(n)     ->     x(n)
2(n)     ->     x(n)x(n)
3(n)     ->     x(n)x(n)x(n)
4(n)     ->     x(n)s(n)
5(n)     ->     s(n)
6(n)     ->     s(n)x(n)
7(n)     ->     s(n)x(n)x(n)
8(n)     ->     s(n)x(n)x(n)x(n)
9(n)     ->     x(n)x(n+1)

x(1)     ->     I
x(2)     ->     X
x(3)     ->     C
x(4)     ->     M
s(1)     ->     V
s(2)     ->     L
s(3)     ->     D
```

2. Develop a program in lex which displays the identifier and the line on which it occurs in the input

file.l

```
%option yylineno

%{
int COMMENT=0;
int id=0;
%}

identifier [a-zA-Z][a-zA-Z0-9]*

%%

"/*" {COMMENT = 1;}
"*/" {COMMENT = 0;}
#.*\n ;

break|case|char|do|double|else|float|for|if|int|return|void|while;



{identifier}\( ;
\( ;
\- ;
\) ;
{identifier}(\[[0-9]*\])? {if(!COMMENT) printf("\n%s IS AN IDENTIFIER AT LINE
NO %d\n",yytext,yylineno);}

\".*\" ;

[0-9]+ ;

\{ ;
\} ;

\++ ;
= ;

\<= |
\>= |
\< |
\== |
```

```
\!= |
\> ;

\, |
\; ;

%%
int main(int argc, char **argv)
{
    FILE *file;
    file=fopen("input.c","r");
    if(!file)
    {
        printf("could not open the file");
        exit(0);
    }
    yyin=file;
    yylex();
    printf("\n");
    return(0);
}

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

Input.c

```
#include <stdio.h>
#define PI 3.14
struct inp
{
    int a;
};
int check(int a, int b)
{
    return (a > b);
}
int main()
{
    struct inp ab;
    int r = 5;
    printf("abc");
    return 0;
}
```

Output :

```
Microsoft Windows [Version 10.0.22000.318]
(c) Microsoft Corporation. All rights reserved.

C:\Users\dangi\OneDrive\Desktop\final practical\practical 2>flex file.l

C:\Users\dangi\OneDrive\Desktop\final practical\practical 2>gcc lex.yy.c

C:\Users\dangi\OneDrive\Desktop\final practical\practical 2>a.exe

struct IS AN IDENTIFIER AT LINE NO 3
inp IS AN IDENTIFIER AT LINE NO 3
a IS AN IDENTIFIER AT LINE NO 5
a IS AN IDENTIFIER AT LINE NO 7
b IS AN IDENTIFIER AT LINE NO 7
a IS AN IDENTIFIER AT LINE NO 9
b IS AN IDENTIFIER AT LINE NO 9
struct IS AN IDENTIFIER AT LINE NO 13
inp IS AN IDENTIFIER AT LINE NO 13
ab IS AN IDENTIFIER AT LINE NO 13
r IS AN IDENTIFIER AT LINE NO 14
```