Roll No: 1803180

Lab Performance Test [2]

Lab Task Q[1]

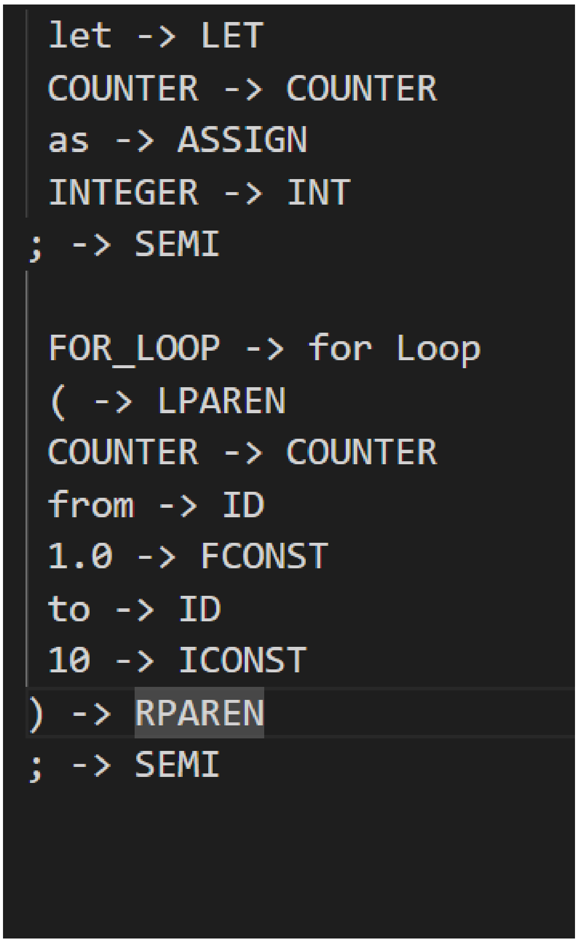
Question: **let COUNTER as INTEGER; FOR_LOOP (COUNTER from 1.0 to 10);**

Solution (Bold your own written code):

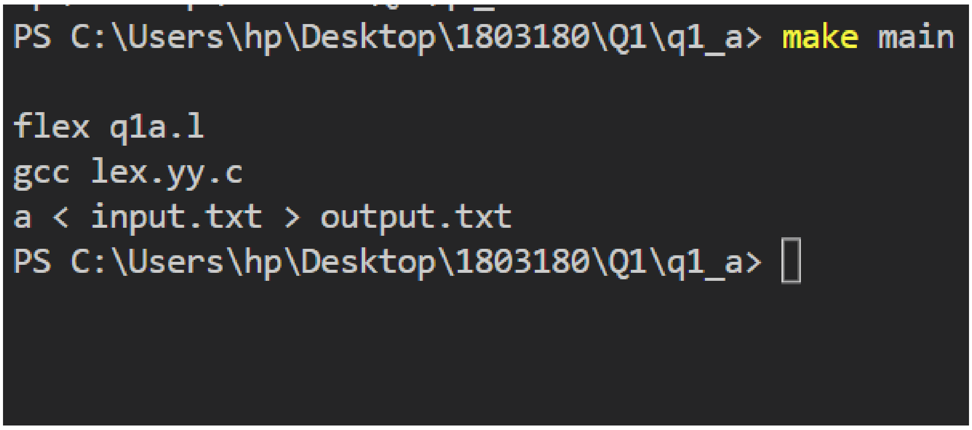
```
%option noyywrap

%{

     #include <stdio.h>
     #include <stdlib.h>
     #include <string.h>
     int lineno = 1; // initialize to 1
     void yyerror();
%}



alpha      [a-zA-Z]
digit      [0-9]
alnum      {alpha}|{digit}
print      [ -~]

ID         {alpha}{alnum}*
ICONST     [0-9]{digit}*
FCONST     {digit}*"."{digit}+
CCONST     (\'{print}\')
STRING    \"{print}*\"




%%
"//".*      { }
"INTEGER"        {printf("%s -> INT\n", yytext);}
"double"       {printf("%s -> DOUBLE\n", yytext);}
"float"        {printf("%s -> FLOAT\n", yytext);}
"char"         {printf("%s -> CHAR\n", yytext);}
"FOR_LOOP"   {printf("%s -> for Loop\n", yytext);}
"input"     {printf("%s -> INPUT\n", yytext);}
"print"     {printf("%s -> PRINT\n", yytext);}
```

```
"let"      {printf("%s -> LET\n", yytext);}
"as"      {printf("%s -> ASSIGN\n", yytext);}
"COUNTER"      {printf("%s -> COUNTER\n", yytext);}

"+"        {printf("%s -> ADDOP\n", yytext);}
"-"        {printf("%s -> SUBOP\n", yytext);}
"*"        {printf("%s -> MULOP\n", yytext);}
"/"        {printf("%s -> DIVOP\n", yytext);}
"++"       {printf("%s -> INCR\n", yytext);}
"--"       {printf("%s -> DECR\n", yytext);}
"||"       {printf("%s -> OROP\n", yytext);}
"&&"       {printf("%s -> ANDOP\n", yytext);}
"!"        {printf("%s -> NOTOP\n", yytext);}
"=="       {printf("%s -> EQUOP\n", yytext);}
"!="       {printf("%s -> NEQUOP\n", yytext);}
">="       {printf("%s -> GTEQ\n", yytext);}
">"        {printf("%s -> GT\n", yytext);}
"<="       {printf("%s -> LTEQ\n", yytext);}
"<"        {printf("%s -> LT\n", yytext);}


"("        {printf("%s -> LPAREN\n", yytext);}
")"        {printf("%s -> RPAREN\n", yytext);}
"{"        {printf("%s -> LBRACE\n", yytext);}
"}"        {printf("%s -> RBRACE\n", yytext);}
":"        {printf("%s -> COLON\n", yytext);}
";"        {printf("%s -> SEMI\n", yytext);}
","        {printf("%s -> COMMA\n", yytext);}
"="        {printf("%s -> ASSIGN\n", yytext);}


{ID}        {printf("%s -> ID\n", yytext);}
{ICONST}    {printf("%s -> ICONST\n", yytext);}
{FCONST}    {printf("%s -> FCONST\n", yytext);}
{CCONST}    {printf("%s -> CCONST\n", yytext);}

%%



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

Output (Screen/SnapShot):



```
 let -> LET
 COUNTER -> COUNTER
 as -> ASSIGN
 INTEGER -> INT
; -> SEMI

 FOR_LOOP -> for Loop
 ( -> LPAREN
 COUNTER -> COUNTER
 from -> ID
 1.0 -> FCONST
 to -> ID
 10 -> ICONST
) -> RPAREN
; -> SEMI
```



```
PS C:\Users\hp\Desktop\1803180\Q1\q1_a> make main

flex q1a.l
gcc lex.yy.c
a < input.txt > output.txt
PS C:\Users\hp\Desktop\1803180\Q1\q1_a> 
```

Question: let COUNTER as INTEGER; FOR_LOOP (COUNTER from 1.0 to 10);

Solution (q1b.l):

```
%option noyywrap

%{



    #include <stdio.h>
    #include <stdlib.h>
    #include <string.h>
    #include "Q1b.tab.h"

    int lineno = 1; // initialize to 1
    void yyerror();
%}

alpha     [a-zA-Z]
digit     [0-9]
alnum     {alpha}|{digit}
print     [ -~]

ID        {alpha}{alnum}*
ICONST    [0-9]{digit}*
FCONST    {digit}*"."{digit}+
CCONST    (\'{print}\')
STRING    \"{print}*\"
%%

"//".*     { }

"INTEGER"      { return INTEGER; }

"FOR_LOOP"      { return FOR_LOOP; }
"COUNTER" { return COUNTER; }
"let"   { return LET; }
"from" { return FROM; }
"to" { return TO; }
"as" { return AS; }
"("        { return LPAREN; }
")"        { return RPAREN; }


{ID}        {return ID;}
{ICONST}    {return ICONST;}
{FCONST}    {return FCONST;}
{CCONST}    {return CCONST;}
```

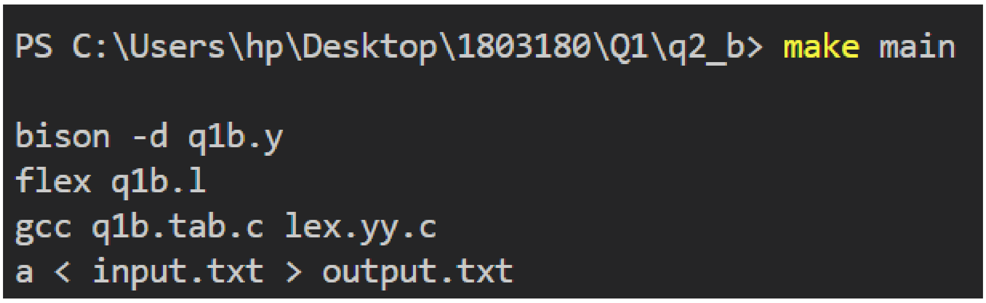
```
"\n"        { lineno += 1; }
[ \t\r\f]+

.           { yyerror("Unrecognized character"); }

%%
```

Solution (q1b.y):

```
%{


#include<stdio.h>
#include <stdlib.h>
#include <string.h>
void yyerror();
extern int lineno;
extern int yylex();

%}

%union
{
    char str_val[00];
    int int_val;
}

%token INT DOUBLE FLOAT CHAR
%token FOR_LOOP AS TYPE COUNTER FROM TO INTEGER LET
%token LPAREN RPAREN LBRACE RBRACE COLON SEMI ASSIGN

%token<str_val> ID
%token ICONST
%token FCONST
%token CCONST

%start STATEMENTS

%%

STATEMENTS: STATEMENTS STATEMENT |
          ;
```

```
STATEMENT: DECLARATION | LOOP
          ;

DECLARATION: LET COUNTER AS INTEGER SEMI
          ;

LOOP: FOR_LOOP COUNT SEMI ;


COUNT: LPAREN COUNTER FROM ICONST TO ICONST RPAREN SEMI;




%%

void yyerror ()
{
     printf("Syntax error at line %d\n", lineno);
     exit(1);
}

int main (int argc, char *argv[])
{
     yyparse();
     printf("Parsing finished!\n");
     return 0;
}
```

Output (Screen/SnapShot):



```
PS C:\Users\hp\Desktop\1803180\Q1\q2_b> make main

bison -d q1b.y
flex q1b.l
gcc q1b.tab.c lex.yy.c
a < input.txt > output.txt
```