Compiler Design Lab

Assignment-6
Implementation of Syntax Checker using Yacc Tool

Name : Vikraman S
Reg No. : 185001195

Code:

analyser.l

```
%{
#include <stdlib.h>
#include <stdio.h>
#include "y.tab.h"
void yyerror(char*);
int linenum=0;
extern int yylval;
%}

%%
\n {linenum++;}
[ \t]+ ;
int|float|char|double {return DATATYPE;}
while {return WHILE;}
if {return IF;}
else {return ELSE;}
for {return FOR;}
[0-9]+ {yylval = atoi(yytext); return INTEGER;}
["].+["] {return STRING;}
[a-zA-Z_]+[0-9]*? {return IDENTIFIER;}
[,=;] {return yytext[0];}
"<"|"<="|">"|">="|"=="|"!=" {return RELATIONAL;}
"+"|"-"|"*"|"/" {return ARITHMETIC;}
"++"|"--" {return UNARYOP;}
"(" {return *yytext;}
")" {return *yytext;}
"{" {return *yytext;}
"}" {return *yytext;}
%%

int yywrap (void)
{
    printf("\nThe given program is syntactically correct.\t\n");
    return 1;
}

void yyerror(char *s)
{
    printf("\nSyntax error at line number %d\n",linenum);
    exit(1);
}
```

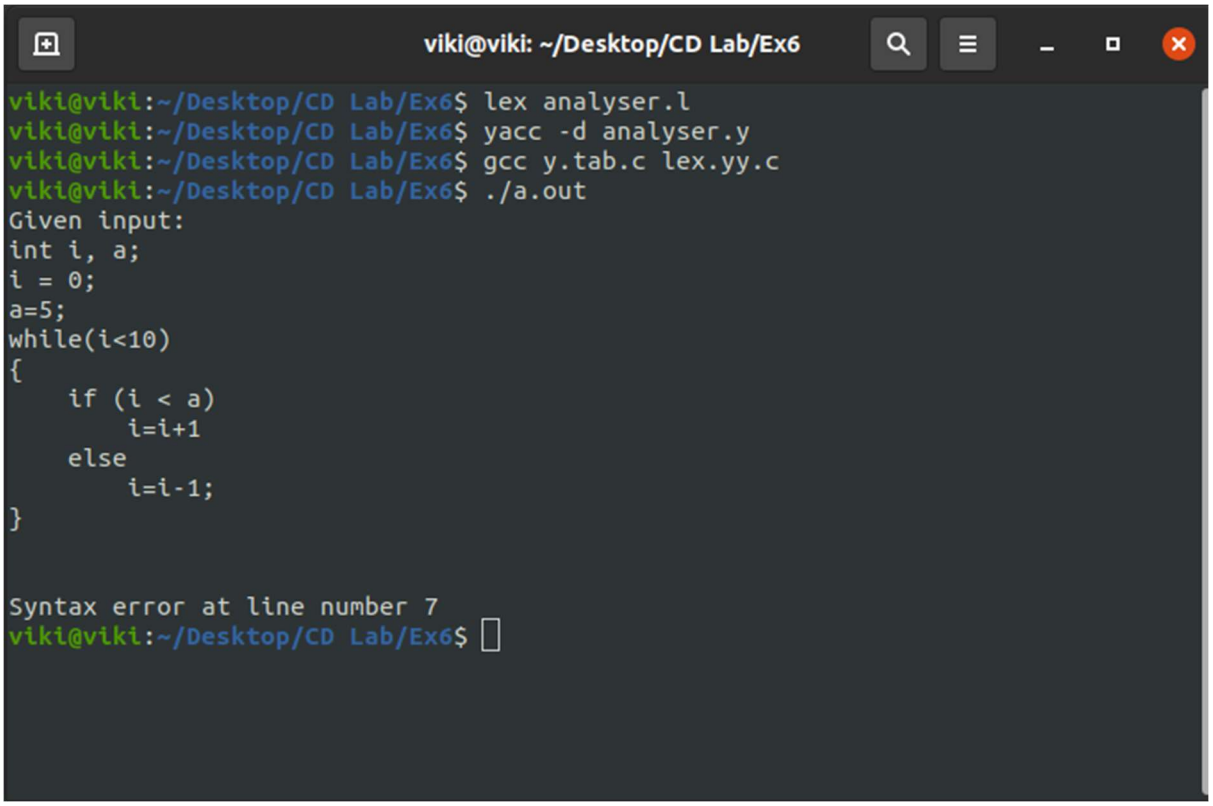
```c
int main()
{
    FILE *fp;
    fp = fopen("input.c","r");
    if (fp == NULL)
    {
        printf("File not found\n");
        exit(0);
    }
    printf("Given input:\n");
    char c =fgetc(fp);
    while(c!=EOF)
    {
        printf("%c",c);
        c=fgetc(fp);
    }
    fclose(fp);
    fp = fopen("input.c","r");
    yyin = fp;
    yyparse();
    fclose(fp);

    return 0;
}
```

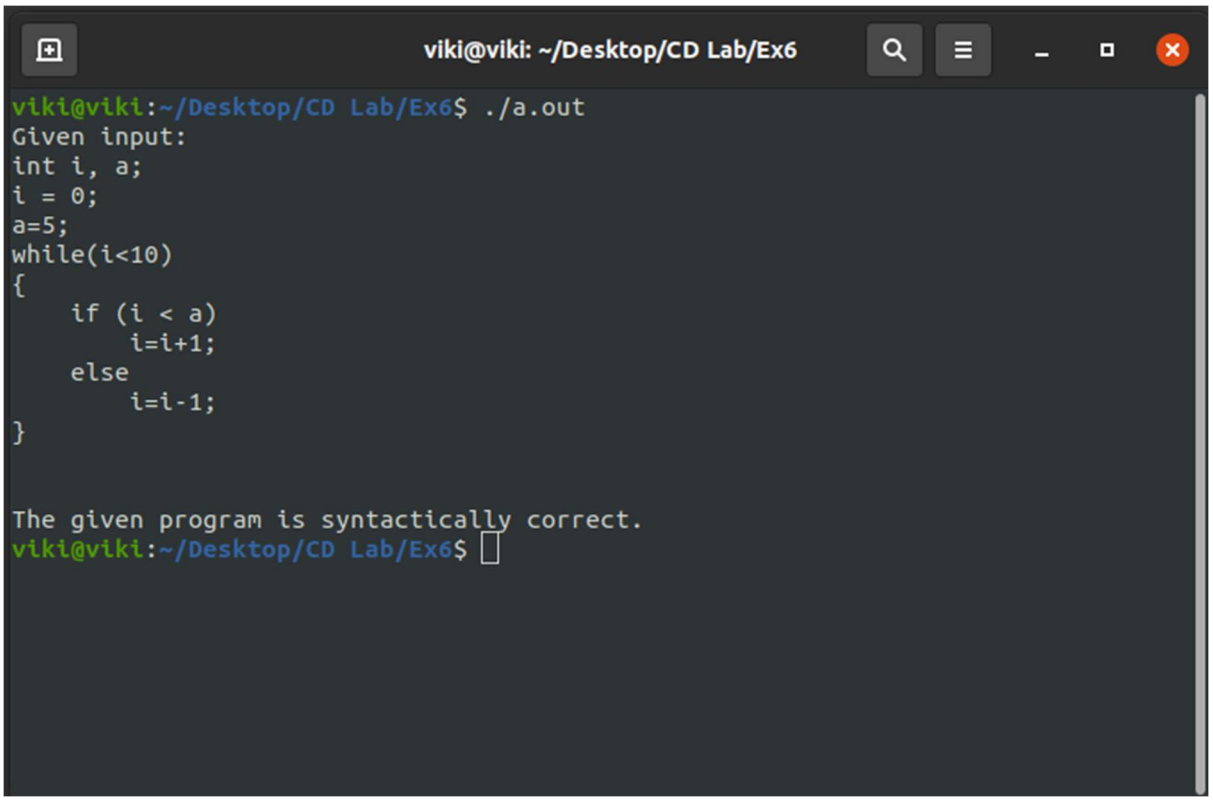
analyser.y

```
%{
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
void yyerror(char*);
int yylex(void);
#include "y.tab.h"
%}

%token INTEGER STRING IDENTIFIER RELATIONAL WHILE FOR ARITHMETIC UNARYOP
DATATYPE IF ELSE

%%
PROGRAM: LINE
LINE: LINE STATEMENT | STATEMENT

STATEMENT: DATATYPE STATEMENTLIST ';'
            | STATEMENTLIST  ';'
            | condition_st

STATEMENTLIST: STATEMENTLIST ',' INIT | INIT

INIT: IDENTIFIER | EXPR

EXPR: IDENTIFIER '=' ASSIGNMENTEXPR | IDENTIFIER UNARYOP | IDENTIFIER '='
ARITHMETIC ASSIGNMENTEXPR
```

```
ASSIGNMENTEXPR: IDENTIFIER ARITHMETIC ASSIGNMENTEXPR
                | INTEGER ARITHMETIC ASSIGNMENTEXPR
                | IDENTIFIER UNARYOP
                | IDENTIFIER
                | INTEGER
                | STRING

condition_st: WHILE '(' condition_expr ')'
                | WHILE '(' condition_expr ')'  '{' LINE  '}'
                | IF '(' condition_expr ')' LINE ELSE
                | IF '(' condition_expr ')'  '{' LINE '}' ELSE  '{' LINE
'}'
                | IF '(' condition_expr ')'   ELSE  '{' LINE  '}'
                | IF '(' condition_expr ')'  '{' LINE '}' ELSE
                | FOR '('DATATYPE IDENTIFIER '='  INTEGER  ';'
condition_expr ';' EXPR ')' '{' LINE  '}'
                | FOR '(' DATATYPE IDENTIFIER '='  INTEGER ';'
condition_expr ';' EXPR ')'

condition_expr: IDENTIFIER RELATIONAL condition_expr
                | INTEGER   RELATIONAL condition_expr
                | IDENTIFIER
                | INTEGER
%%
```

Output:




viki@viki: ~/Desktop/CD Lab/Ex6
viki@viki:~/Desktop/CD Lab/Ex6$ ./a.out
Given input:
int i, a;
i = 0;
a=5;
while(i<10)
{
if (i < a)
i=i+1;
else
i=i-1;
}
The given program is syntactically correct.
viki@viki:~/Desktop/CD Lab/Ex6$