SSN COLLEGE OF ENGINEERING, KALAVAKKAM

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

UCS1602 - Compiler Design Programming

Assignment-6

Syntax checker using LEX and YACC

Name : Rahul Ram M

Reg no. : 185001121

CODE:

checker.l:

```
%{
#include <stdlib.h>
#include <stdio.h>
#include "y.tab.h"
void yyerror(char *s);
extern int yylval;
int lno = 0;
char* t = "test.c";
%}
%%
[ \t]+                          ;
\n                              {lno++;}
int|float|char|double           {return DATATYPE;}
while                           {return WHILE;}
if                              {return IF;}
else                            {return ELSE;}
```

```
for                                     {return FOR;}

[0-9]+                                  {yylval = atoi(yytext); return NUMBER;}
[0-9]+\.*[0-9]*                 {yylval = atoi(yytext); return NUMBER;}
["].+["]                                {return STRING;}
[a-zA-Z_]+[0-9]*?                       {return VARIABLE;}
[,=;]                                   {return yytext[0];}
"<"|"<="|">"|">="|"=="|"!="             {return RELATIONAL;}
"+"|"-"|"*"|"/"                         {return ARITH;}
"++"|"--"                               {return UNARY;}
"("                                     {return *yytext;}
")"                                     {return *yytext;}
"{"                                     {return *yytext;}
"}"                                     {return *yytext;}
.                                       {yyerror(yytext);}
%%
int yywrap (void) {
return 1;
}

void yyerror(char *s)
{
printf("%s:line:%d:syntax error\n",t,lno);
exit(1);
}

int main()
```

```
{
FILE *fp;
fp = fopen(t,"r");
if (fp == NULL)
{
printf("File not found error\n");
exit(0);
}
yyin = fp;
yyparse();
printf("Syntax Correct!\n");
fclose(fp);
return 0;
}
```

checker.y:

```
%{
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
int yylex(void);
#include "y.tab.h"
%}
%token NUMBER STRING VARIABLE RELATIONAL WHILE FOR ARITH UNARY DATATYPE IF ELSE
%%
PROGRAM:LINE
LINE :LINE STATEMENT| STATEMENT
```

STATEMENT: DATATYPE STATEMENT_LIST ';'

|STATEMENT_LIST ';'

|CONDITION_ST

STATEMENT_LIST : STATEMENT_LIST ',' VAR | VAR

VAR :VARIABLE | EXPR

EXPR:VARIABLE '=' ASSIGNMENT_EXP | VARIABLE UNARY | VARIABLE '=' ARITH ASSIGNMENT_EXP

ASSIGNMENT_EXP : VARIABLE ARITH ASSIGNMENT_EXP

|NUMBER ARITH ASSIGNMENT_EXP

|VARIABLE UNARY

|VARIABLE

| NUMBER

| STRING

CONDITION_ST : WHILE '(' CONDITION_EXP ')'

|WHILE '(' CONDITION_EXP ')' '{' LINE '}'

|IF '(' CONDITION_EXP ')' LINE ELSE LINE

|IF '(' CONDITION_EXP ')' '{' LINE '}' ELSE '{' LINE '}'

|IF '(' CONDITION_EXP ')' '{' '}' ELSE '{' LINE '}'

|IF '(' CONDITION_EXP ')' '{' LINE '}'

|IF '(' CONDITION_EXP ')' LINE

|FOR '('DATATYPE VARIABLE '=' NUMBER ';' CONDITION_EXP ';' EXPR ')' '{' LINE '}'

|FOR '(' DATATYPE VARIABLE '=' NUMBER ';' CONDITION_EXP ';' EXPR ')'

|FOR '(' VARIABLE '=' NUMBER ';' CONDITION_EXP ';' EXPR ')'

CONDITION_EXP : VARIABLE RELATIONAL CONDITION_EXP

| NUMBER RELATIONAL CONDITION_EXP

```
 | VARIABLE
 | NUMBER
%%
```

Sample I/O:

Without syntax error:

test.c:

```
int a = 10, b;
b = 5;
for(int j = 0; j < 5; j++)
{
b=b+1;
}
while(b < 20)
{
b++;
}
```

Output:

Syntax Correct!

With Syntax error:

test.c:

```
int a = 10, b;
b = 5;
for(int j = 0; j < 5; j++)
{
b=b+1;
}
```

```
while(b < 20)
{
b++
}
```

Output:

```
test.c:line:9:syntax error
```

Learning Outcomes:

This assignment helped me to

1. Understand the implementation of syntax checker using LEX and YACC.
2. Understand the Grammar used to check the syntax.