Sharvan Ram Kumaran
CSE-C
185001143
Assignment 6

Develop a Syntax checker to recognize the tokens necessary for the following statements by writing suitable grammars

Assignment statement

Conditional statement

Looping statement

CHECK.L

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

assign     ("=")
relop      ("=="|"!="|">="|"<="|"<"|">")
arithop    ("+"|"-"|"/"|"%"|"*")
inde     ("++"|"--")
logical    ("||"|"&&")
id  [a-zA-Z_][a-zA-Z0-9_]*


%%

[0-9]+        {return NUM;}
{assign}       {return ASSIGN;}
{relop}        {return RELOP;}
{logical}      {return LOGIC;}
{arithop}      {return ARITH;}
{inde}         {return INDE;}
"if"          {return IF;}
"else if"      {return ELSEIF;}
"else"         {return ELSE;}
"for"         {return FOR;}
```

```
"while"        {return WHILE;}
{id}               {return ID;}


[ \t]          {;}
[\n]           {;}
.             {return *yytext;}

%%

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

CHECK.Y

```
%{
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
int yylex(void);
int yyerror(char *);
int flag=0;
%}


%token NUM ASSIGN ID
%token IF ELSEIF ELSE
%token FOR WHILE
%token RELOP LOGIC ARITH INDE

%%

stmts   :   bl stmts
        |   bl
        ;

bl      :   Loop '{' bl
        |   condstmt '{' bl
        |   expression ';'
        |   '}'
        ;

Loop    :   FOR '(' expression ';' condition ';' expression ')'
```

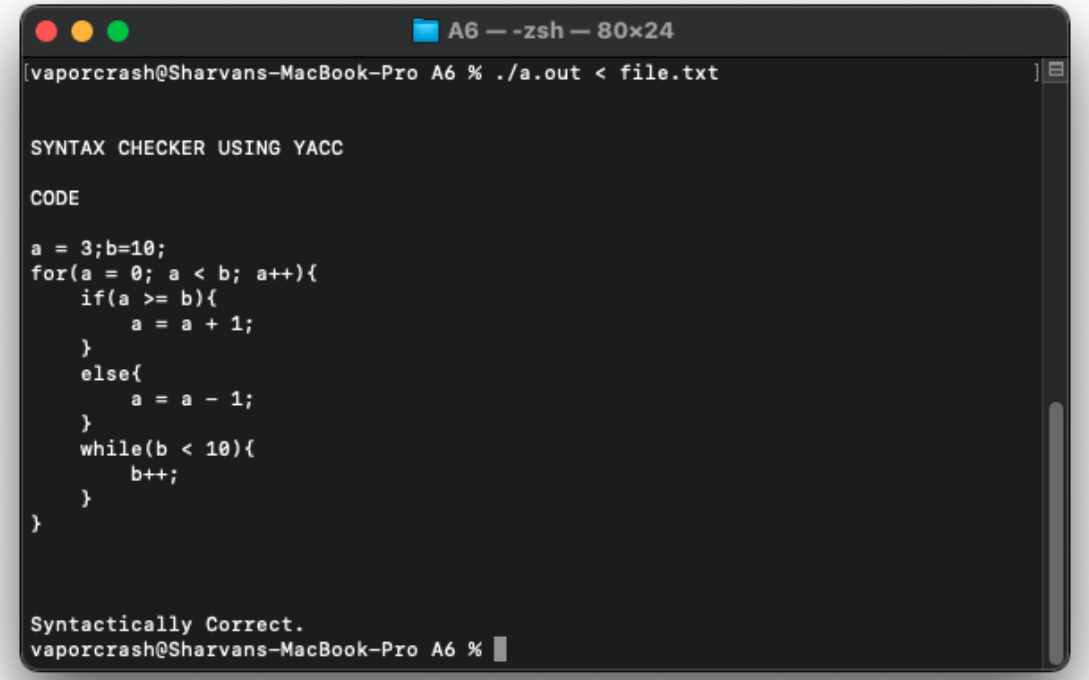
```
      |   FOR '(' ';' condition ';' ')'
      |   WHILE '(' condition ')'
      ;

condstmt :   IF '(' condition ')'
       |   ELSEIF '(' condition ')'
       |   ELSE
       ;

condition       :   condn LOGIC condition
          |   condn
                ;

condn     :   ID RELOP ID
            |   ID RELOP NUM
            |   ID
            ;

expression   :   init
                    |   ID ASSIGN ID ARITH ID
                    |   ID ASSIGN ID ARITH NUM
                    |   ID ASSIGN NUM ARITH NUM
                    |   ID INDE
                    |   INDE ID
                    ;

init   :   ID ASSIGN init
      |   ID ASSIGN ID
      |   ID ASSIGN NUM
      ;
%%

int yyerror(char *s)
{
   flag = 1;
   return 1;
}

int main(void)
{
   printf("\n\nSYNTAX CHECKER USING YACC\n");
   printf("\nCODE\n\n");
   FILE *fp = fopen("file.txt", "r+");
   char c = fgetc(fp);
```

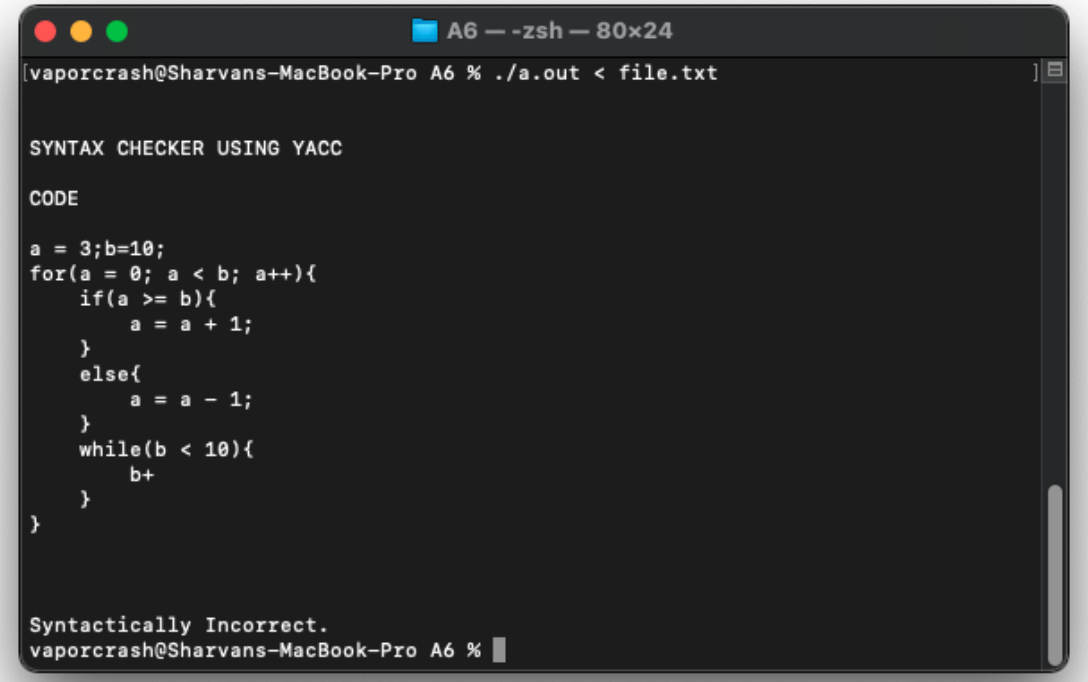
```
    while (c != EOF)
    {
        printf ("%c", c);
        c = fgetc(fp);
    }
    fclose(fp);
    printf("\n\n");
    FILE *fps = fopen("file.txt", "r+");
    yyparse();
    fclose(fps);
    if(flag==1)
    {
        printf("\nSyntactically Incorrect.\n");
    }

    else
    {
        printf("\nSyntactically Correct.\n");
    }

    return 0;
}
```

FILE.TXT

```
a = 3;b=10;
for(a = 0; a < b; a++){
    if(a >= b){
        a = a + 1;
    }
    else{
        a = a - 1;
    }
    while(b < 10){
        b++;
    }
}
```

OUTPUT:


A6 — -zsh — 80×24
vaporcrash@Sharvans-MacBook-Pro A6 % ./a.out < file.txt
SYNTAX CHECKER USING YACC
CODE
a = 3;b=10;
for(a = 0; a < b; a++){
if(a >= b){
a = a + 1;
}
else{
a = a - 1;
}
while(b < 10){
b++;
}
}
Syntactically Correct.
vaporcrash@Sharvans-MacBook-Pro A6 %



LEARNING OBJECTIVE:

Syntax Checker was implemented using yacc.