IT250 – AUTOMATA & COMPILER DESIGN

ASSIGNMENT 6

Name: **Sachin Prasanna**
Roll No.: **211IT058**

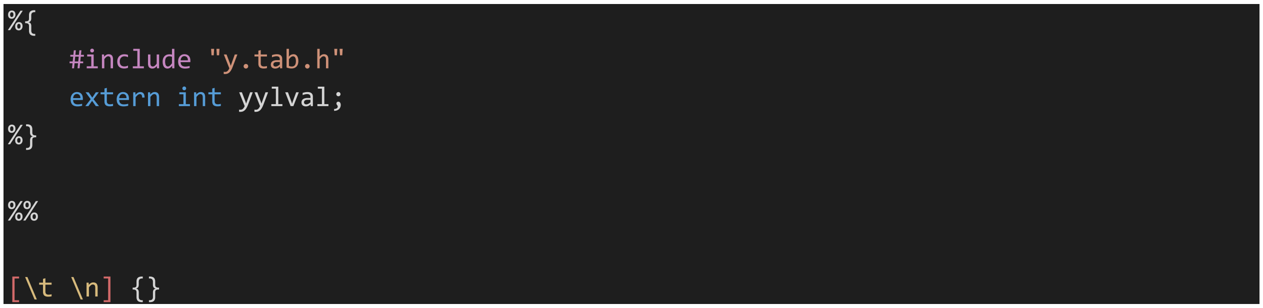
Notations:

$ - Signifies code is completed and tells the parser to check the code if correct.

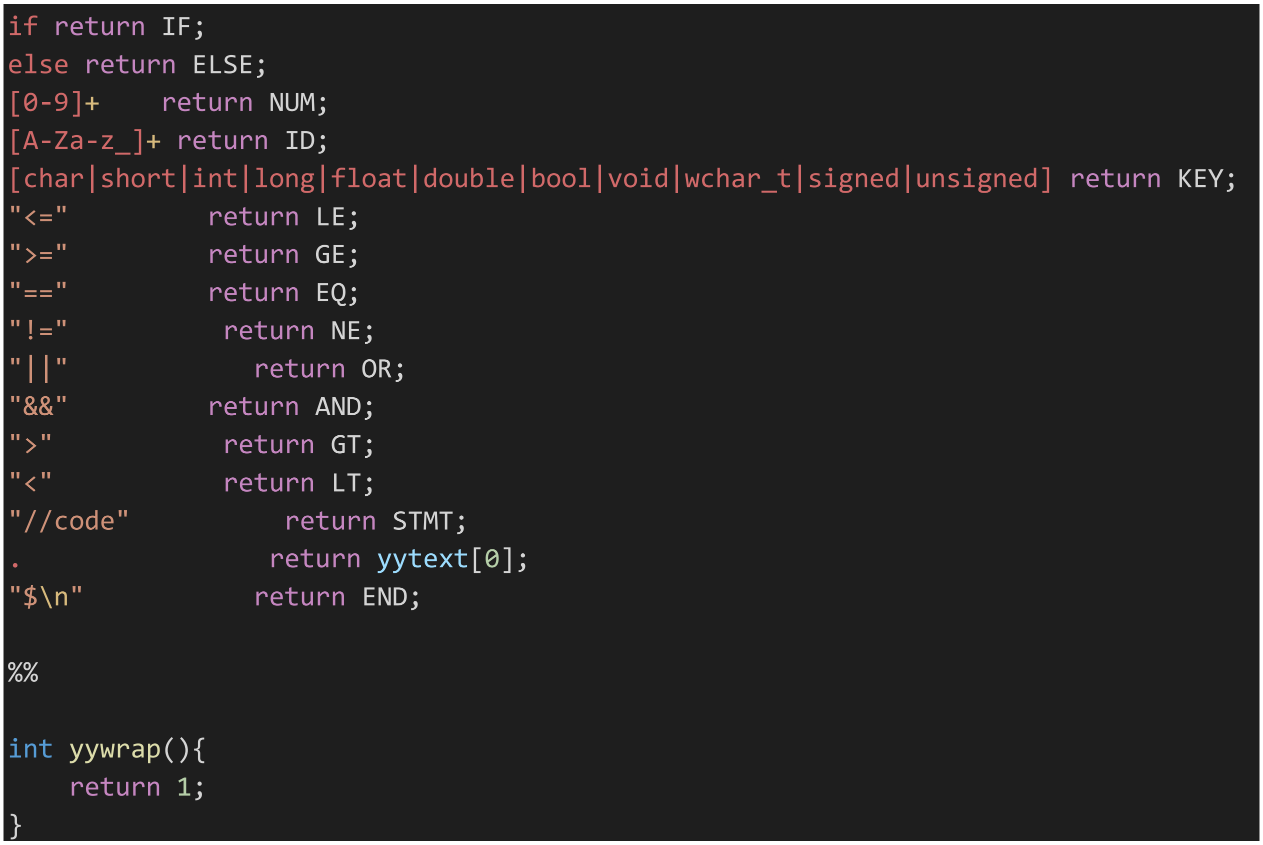
//code – Signifies it is some valid code written in the C programming language.

1a)

LEX Code:

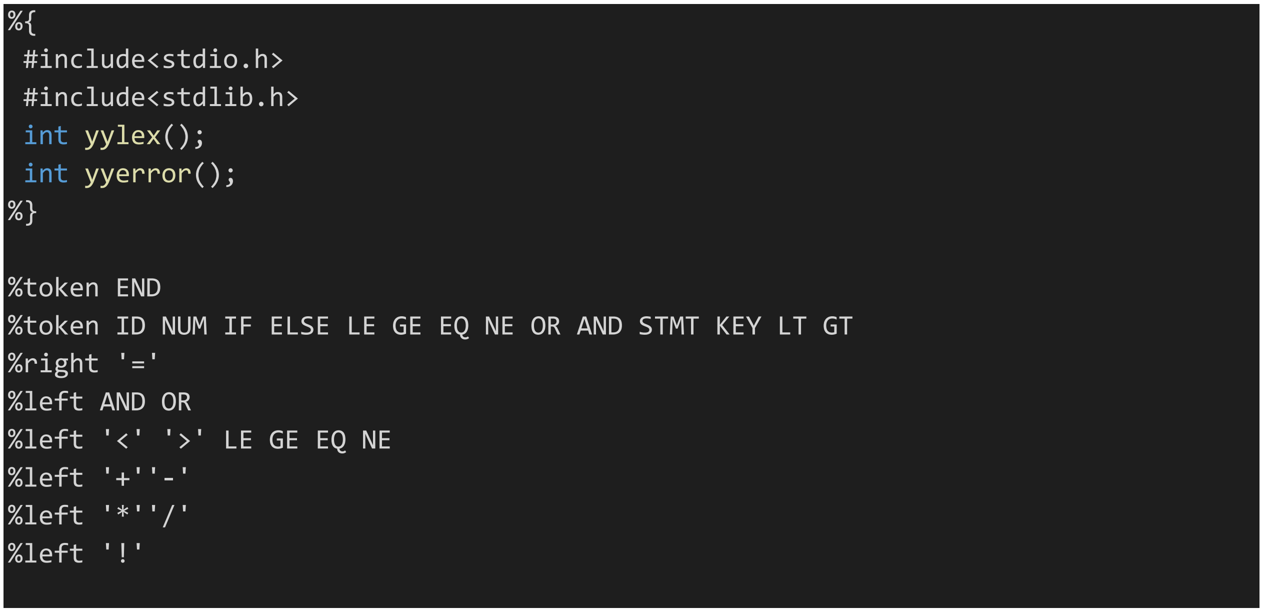


```
%{
    #include "y.tab.h"
    extern int yylval;
%}

%%

[\t \n] {}
```

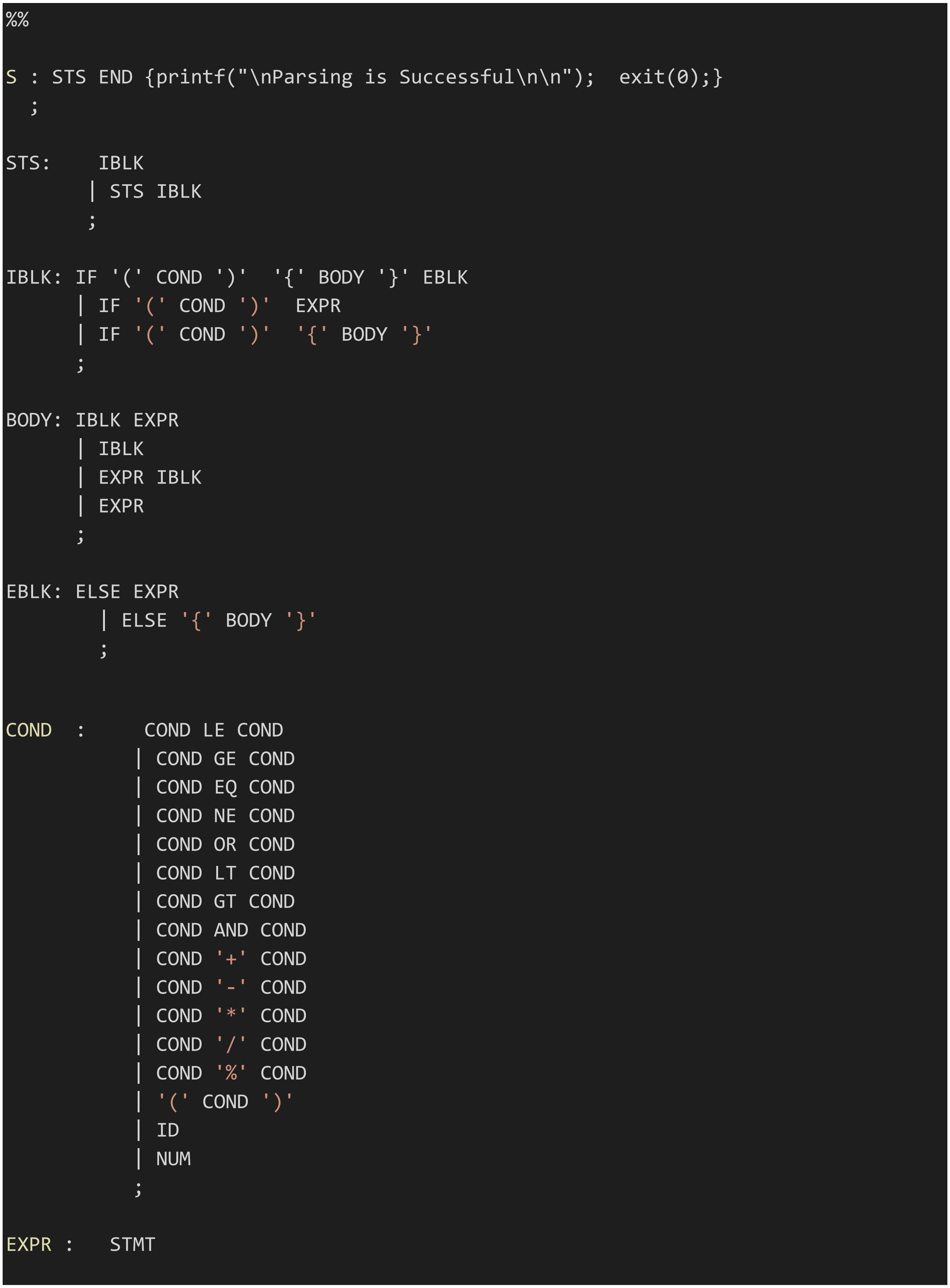


```
if return IF;
else return ELSE;
[0-9]+    return NUM;
[A-Za-z_]+ return ID;
[char|short|int|long|float|double|bool|void|wchar_t|signed|unsigned] return KEY;
"<="         return LE;
">="         return GE;
"=="         return EQ;
"!="          return NE;
"||"            return OR;
"&&"         return AND;
">"           return GT;
"<"           return LT;
"//code"          return STMT;
.                return yytext[0];
"$\n"           return END;

%%

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

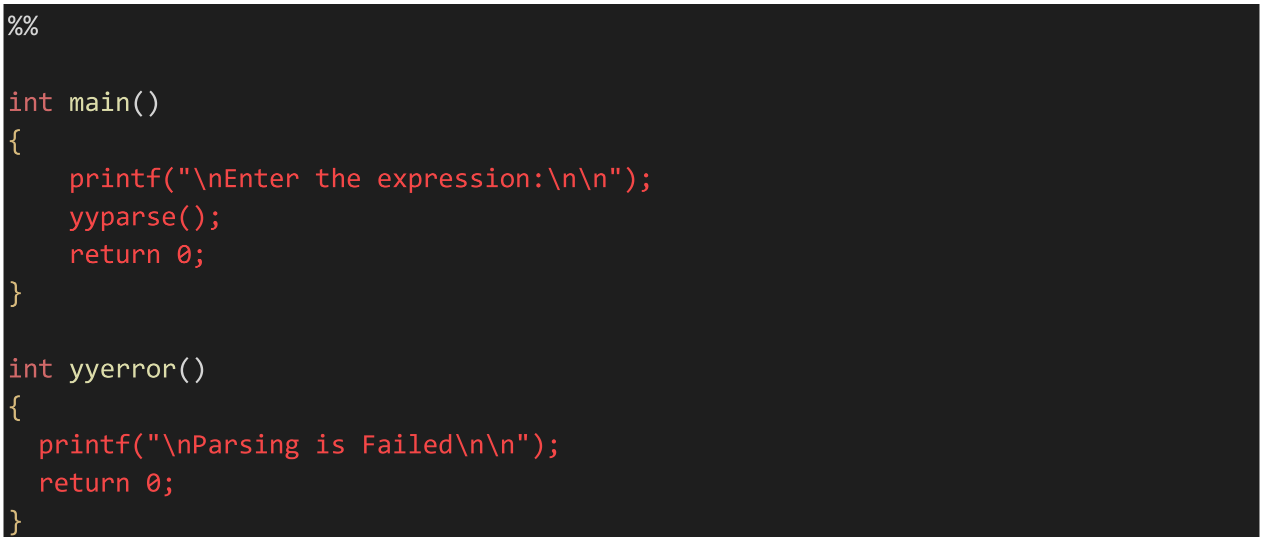
YACC Code:



```
%{
 #include<stdio.h>
 #include<stdlib.h>
 int yylex();
 int yyerror();
%}

%token END
%token ID NUM IF ELSE LE GE EQ NE OR AND STMT KEY LT GT
%right '='
%left AND OR
%left '<' '>' LE GE EQ NE
%left '+''-'
%left '*''/'
%left '!'
```

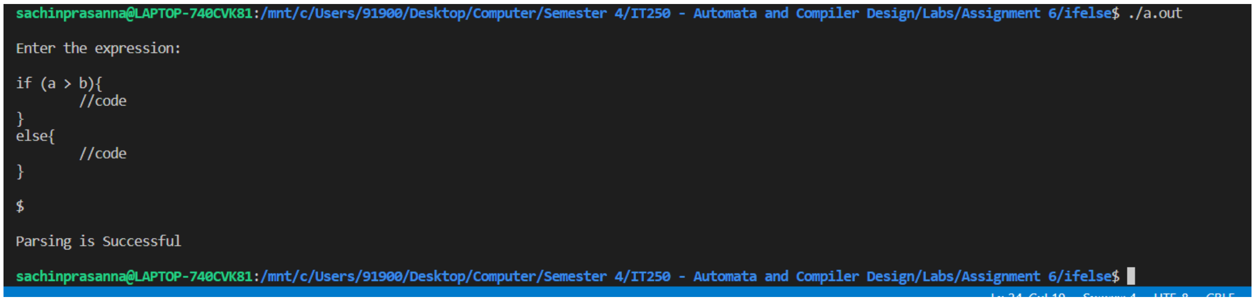


```
%%

S : STS END {printf("\nParsing is Successful\n\n");  exit(0);}
  ;

STS:    IBLK
       | STS IBLK
       ;

IBLK: IF '(' COND ')'  '{' BODY '}' EBLK
      | IF '(' COND ')'  EXPR
      | IF '(' COND ')'  '{' BODY '}'
      ;

BODY: IBLK EXPR
      | IBLK
      | EXPR IBLK
      | EXPR
      ;

EBLK: ELSE EXPR
        | ELSE '{' BODY '}'
        ;


COND  :     COND LE COND
           | COND GE COND
           | COND EQ COND
           | COND NE COND
           | COND OR COND
           | COND LT COND
           | COND GT COND
           | COND AND COND
           | COND '+' COND
           | COND '-' COND
           | COND '*' COND
           | COND '/' COND
           | COND '%' COND
           | '(' COND ')'
           | ID
           | NUM
           ;

EXPR :   STMT
```

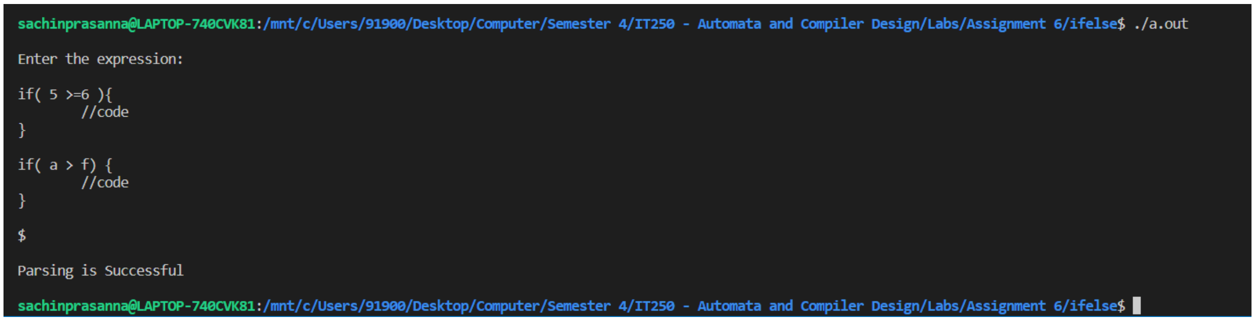


```
%%

int main()
{
    printf("\nEnter the expression:\n\n");
    yyparse();
    return 0;
}

int yyerror()
{
  printf("\nParsing is Failed\n\n");
  return 0;
}
```

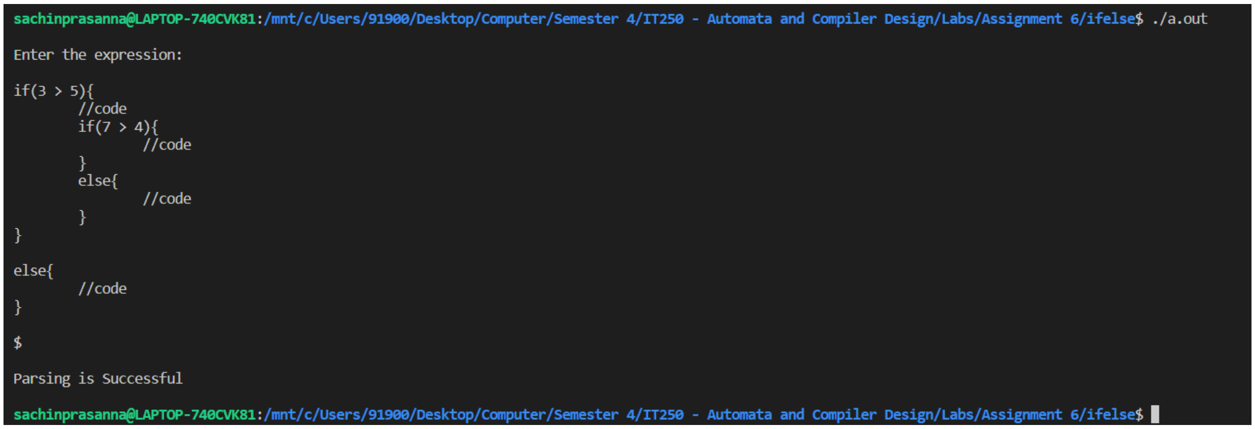
Outputs:



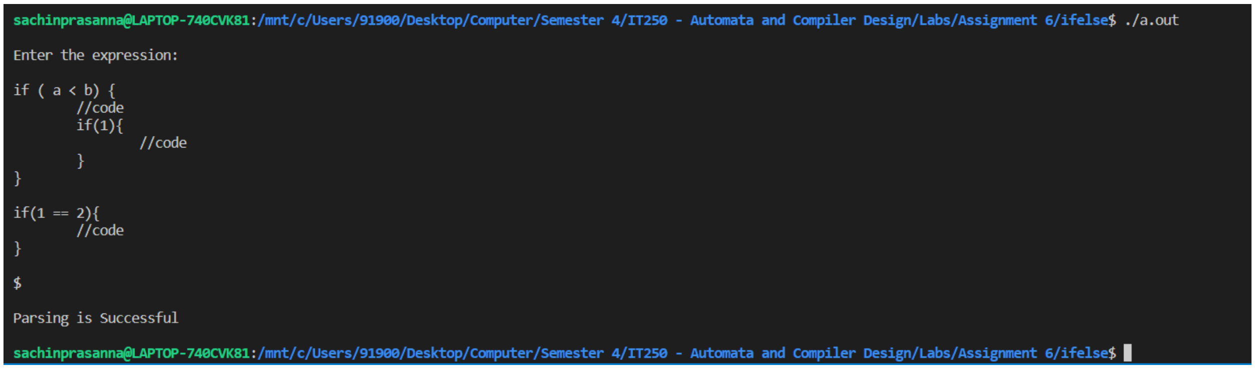
```
sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6/ifelse$ ./a.out

Enter the expression:

if (a > b){
        //code
}
else{
        //code
}

$

Parsing is Successful

sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6/ifelse$
```



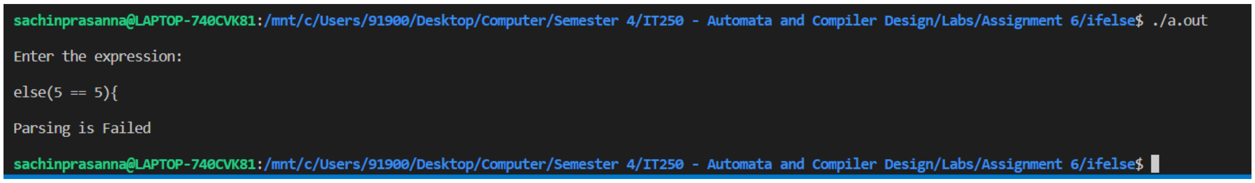
```
sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6/ifelse$ ./a.out

Enter the expression:

if( 5 >=6 ){
        //code
}

if( a > f) {
        //code
}

$

Parsing is Successful

sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6/ifelse$
```



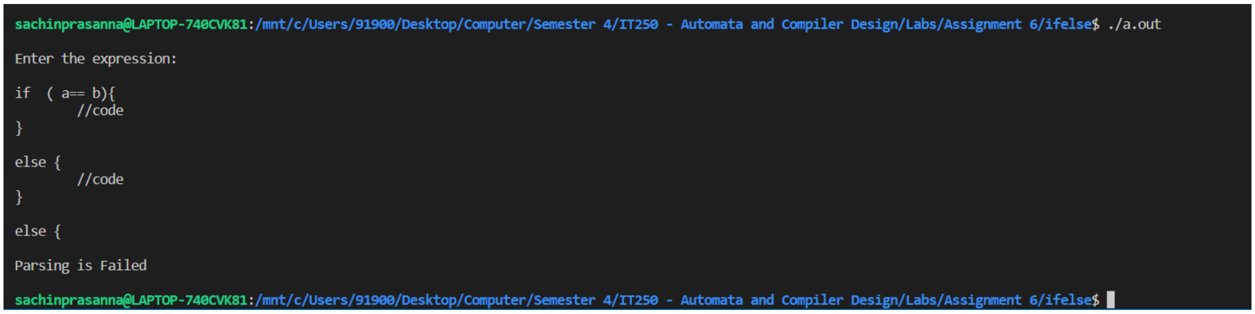
```
sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6/ifelse$ ./a.out

Enter the expression:

if(3 > 5){
        //code
        if(7 > 4){
                //code
        }
        else{
                //code
        }
}

else{
        //code
}

$

Parsing is Successful

sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6/ifelse$
```



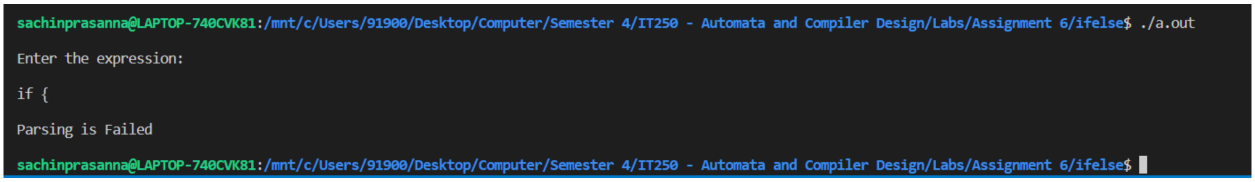
```
sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6/ifelse$ ./a.out

Enter the expression:

if ( a < b) {
        //code
        if(1){
                //code
        }
}

if(1 == 2){
        //code
}

$

Parsing is Successful

sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6/ifelse$
```



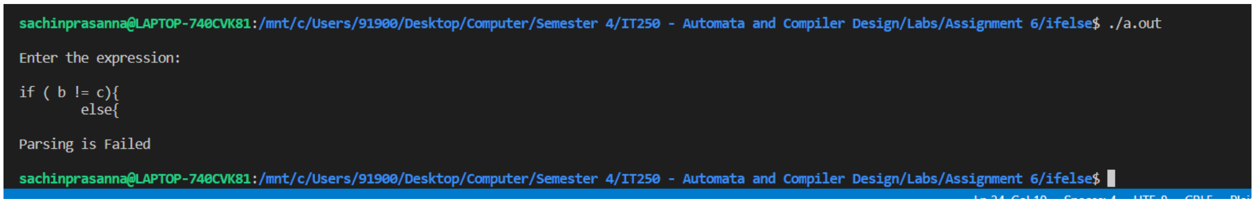
```
sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6/ifelse$ ./a.out

Enter the expression:

else(5 == 5){

Parsing is Failed

sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6/ifelse$
```



```
sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6/ifelse$ ./a.out

Enter the expression:

if  ( a== b){
        //code
}

else {
        //code
}

else {

Parsing is Failed

sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6/ifelse$
```



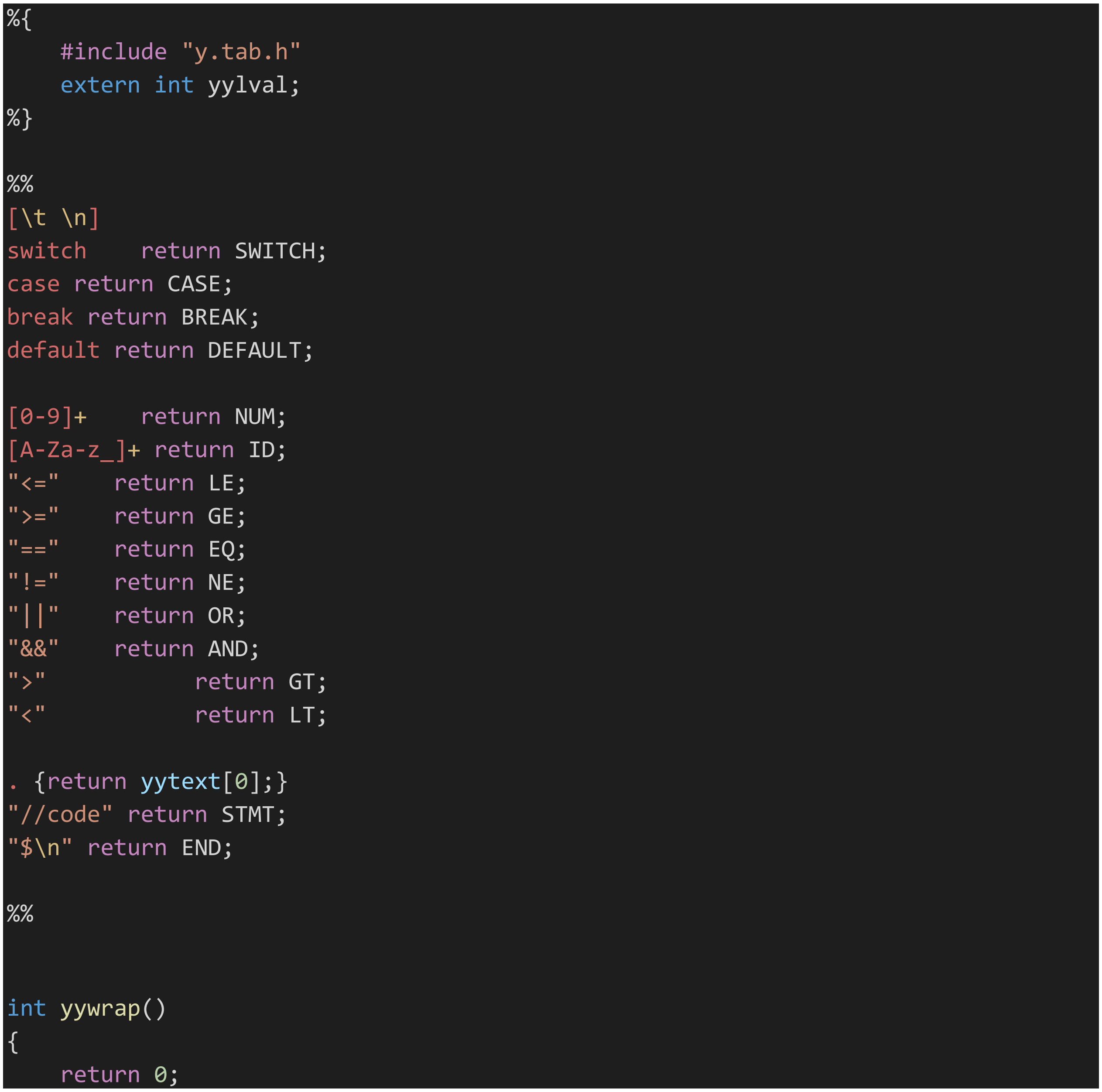
```
sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6/ifelse$ ./a.out

Enter the expression:

if {

Parsing is Failed

sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6/ifelse$
```



```
sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6/ifelse$ ./a.out

Enter the expression:

if ( b != c){
        else{

Parsing is Failed

sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6/ifelse$
```

1b)

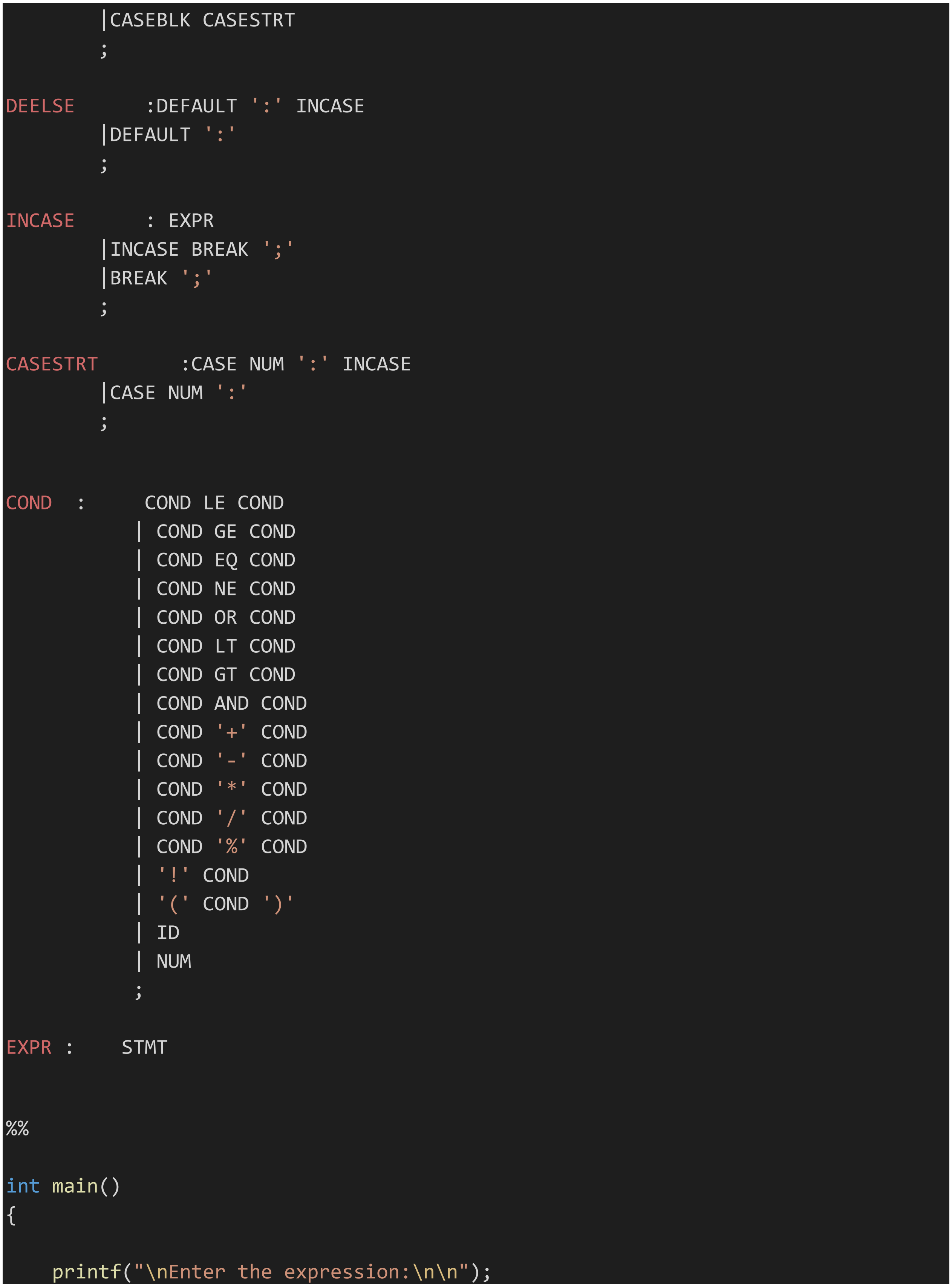
LEX Code:



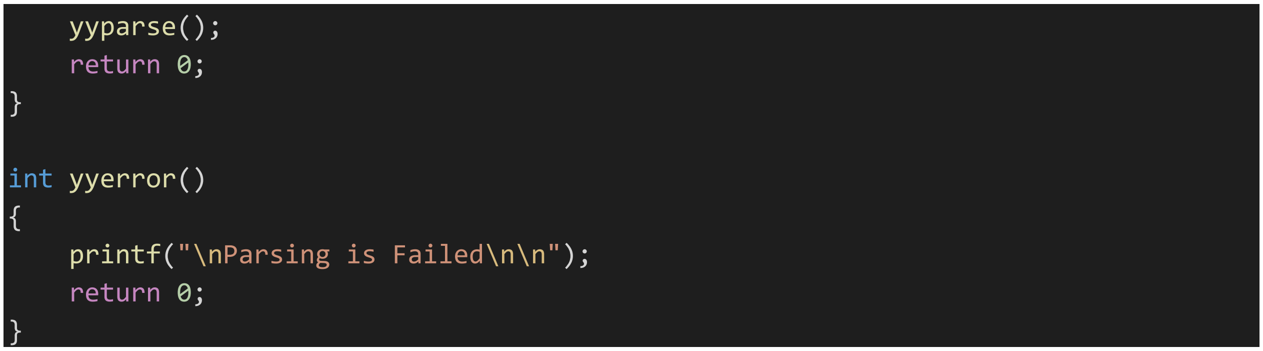
```
%{
    #include "y.tab.h"
    extern int yylval;
%}

%%
[\t \n]
switch    return SWITCH;
case return CASE;
break return BREAK;
default return DEFAULT;

[0-9]+    return NUM;
[A-Za-z_]+ return ID;
"<="    return LE;
">="    return GE;
"=="    return EQ;
"!="    return NE;
"||"    return OR;
"&&"    return AND;
">"           return GT;
"<"           return LT;

. {return yytext[0];}
"//code" return STMT;
"$\n" return END;

%%


int yywrap()
{
    return 0;
}
```

YACC Code:

```
%{
    #include <stdio.h>
    #include <stdlib.h>
    int yylex();
    int yyerror();
%}

%token END
%token ID NUM SWITCH CASE DEFAULT BREAK LE GE EQ NE OR AND STMT LT GT
%right '='
%left AND OR
%left '<' '>' LE GE EQ NE
%left '+''-'
%left '*''/'
%left '!'

%%

S       :SWITCHSTMT END {printf("\nParsing is Successful\n\n"); exit (0);}
        ;

SWITCHSTMT      :STTS '{' BODY '}'
        |SWITCHSTMT STTS '{' BODY '}'
        ;

BODY    : CASEBLK
        | CASEBLKD
        ;


STTS      :SWITCH '(' COND ')'
        ;

CASEBLKD     :CASEBLK DEELSE
        |CASEBLKD CASESTRT
        |DEELSE
        ;

CASEBLK      :CASESTRT
```

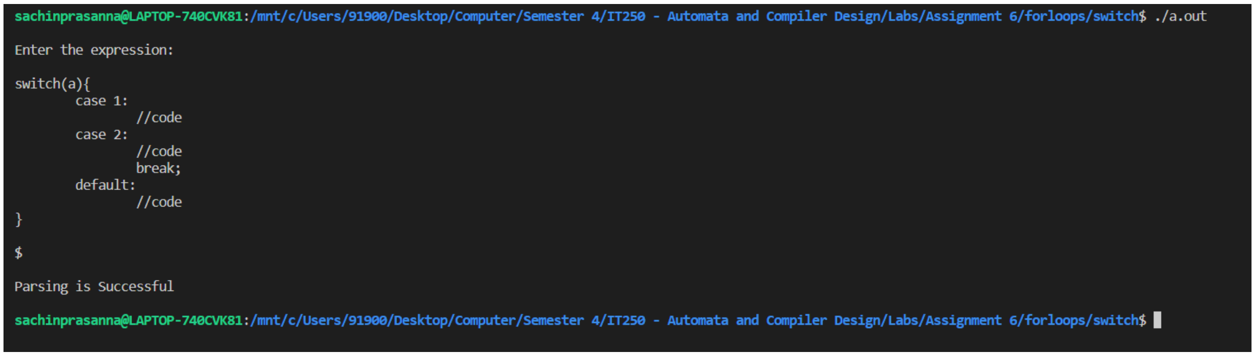


```
        |CASEBLK CASESTRT
        ;

DEELSE      :DEFAULT ':' INCASE
        |DEFAULT ':'
        ;

INCASE      : EXPR
        |INCASE BREAK ';'
        |BREAK ';'
        ;

CASESTRT       :CASE NUM ':' INCASE
        |CASE NUM ':'
        ;


COND  :     COND LE COND
           | COND GE COND
           | COND EQ COND
           | COND NE COND
           | COND OR COND
           | COND LT COND
           | COND GT COND
           | COND AND COND
           | COND '+' COND
           | COND '-' COND
           | COND '*' COND
           | COND '/' COND
           | COND '%' COND
           | '!' COND
           | '(' COND ')'
           | ID
           | NUM
           ;

EXPR :    STMT


%%

int main()
{

    printf("\nEnter the expression:\n\n");
```

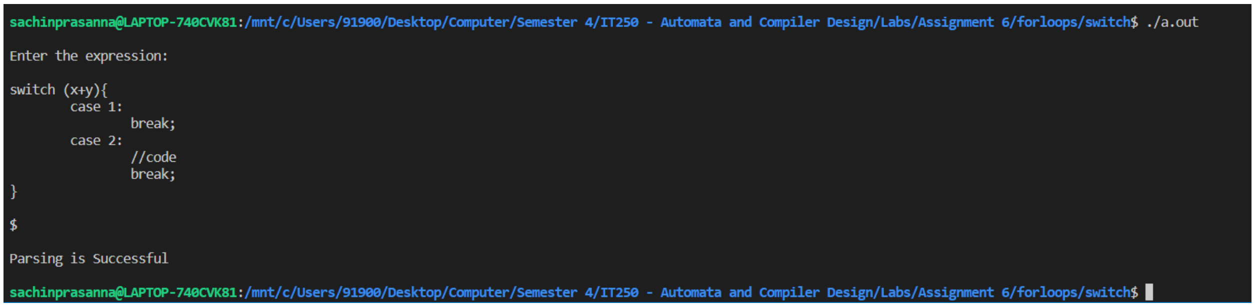


```
    yyparse();
    return 0;
}

int yyerror()
{
    printf("\nParsing is Failed\n\n");
    return 0;
}
```

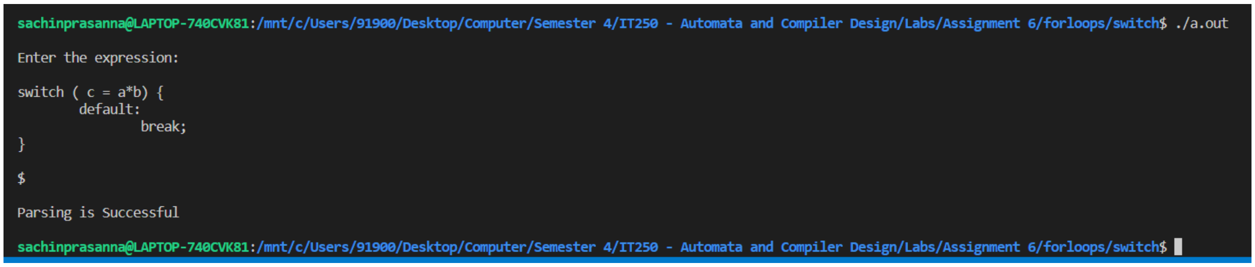
Outputs:



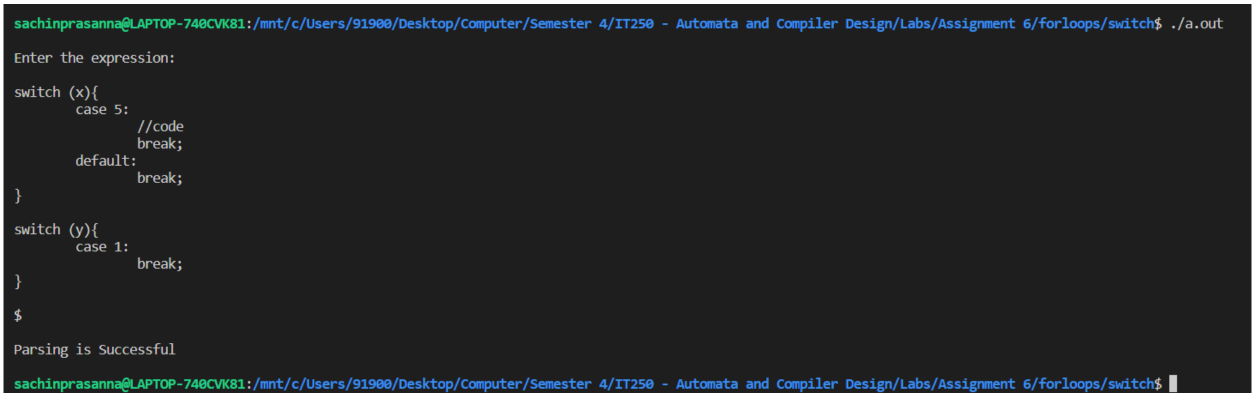
```
sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6/forloops/switch$ ./a.out

Enter the expression:

switch(a){
        case 1:
                //code
        case 2:
                //code
                break;
        default:
                //code
}

$

Parsing is Successful

sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6/forloops/switch$
```



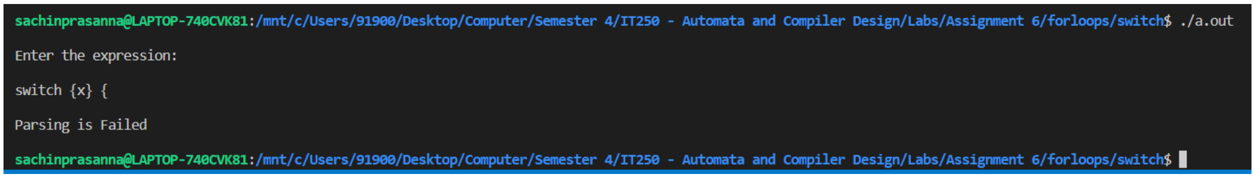
```
sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6/forloops/switch$ ./a.out

Enter the expression:

switch (x+y){
        case 1:
                break;
        case 2:
                //code
                break;
}

$

Parsing is Successful

sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6/forloops/switch$
```



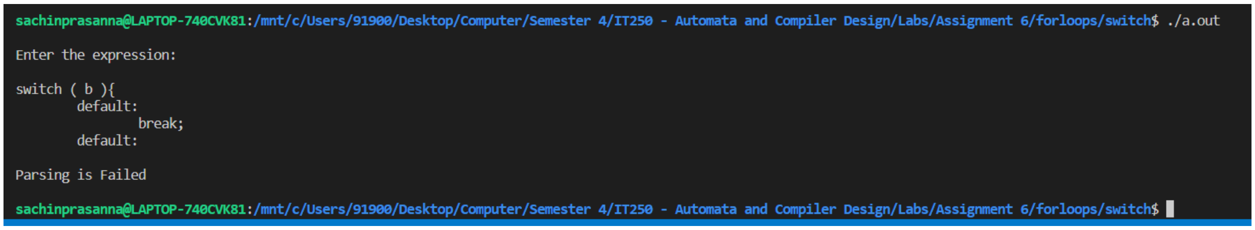
```
sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6/forloops/switch$ ./a.out

Enter the expression:

switch ( c = a*b) {
        default:
                break;
}

$

Parsing is Successful

sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6/forloops/switch$
```



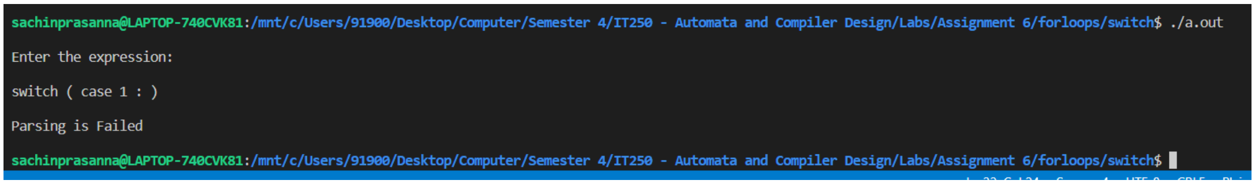
```
sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6/forloops/switch$ ./a.out
Enter the expression:
switch (x){
        case 5:
                //code
                break;
        default:
                break;
}
switch (y){
        case 1:
                break;
}
$
Parsing is Successful
sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6/forloops/switch$
```



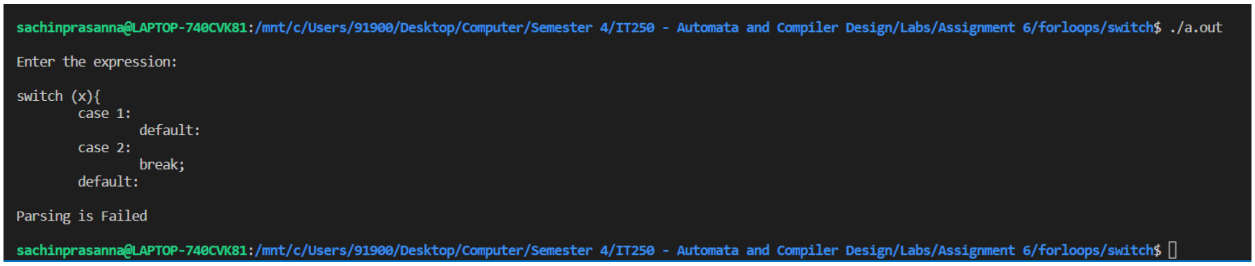
```
sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6/forloops/switch$ ./a.out
Enter the expression:
switch {x} {
Parsing is Failed
sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6/forloops/switch$
```



```
sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6/forloops/switch$ ./a.out
Enter the expression:
switch ( b ){
        default:
                break;
        default:
Parsing is Failed
sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6/forloops/switch$
```



```
sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6/forloops/switch$ ./a.out
Enter the expression:
switch ( case 1 : )
Parsing is Failed
sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6/forloops/switch$
```



```
sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6/forloops/switch$ ./a.out
Enter the expression:
switch (x){
        case 1:
                default:
        case 2:
                break;
        default:
Parsing is Failed
sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6/forloops/switch$
```

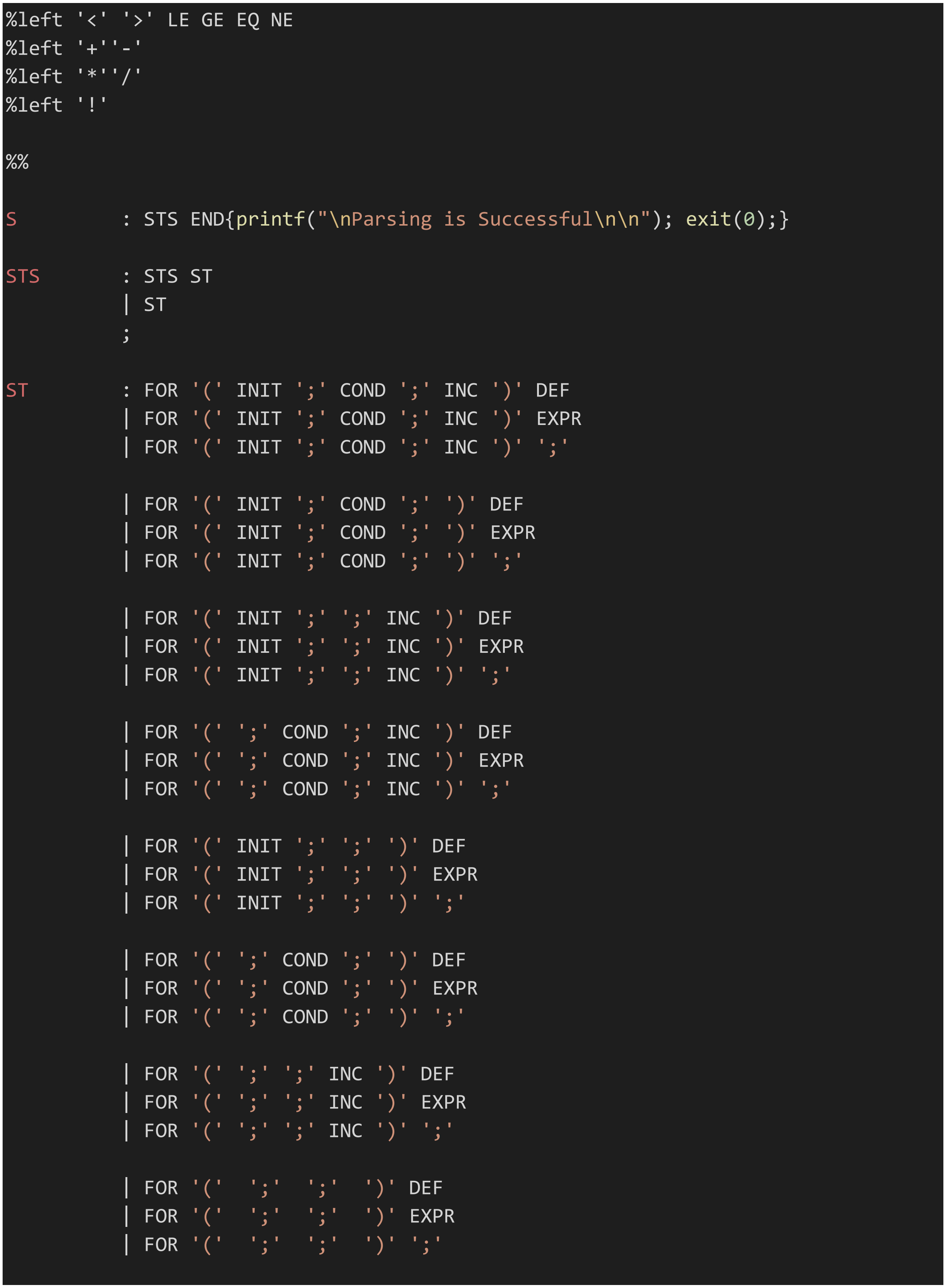
2 a)

LEX Code:

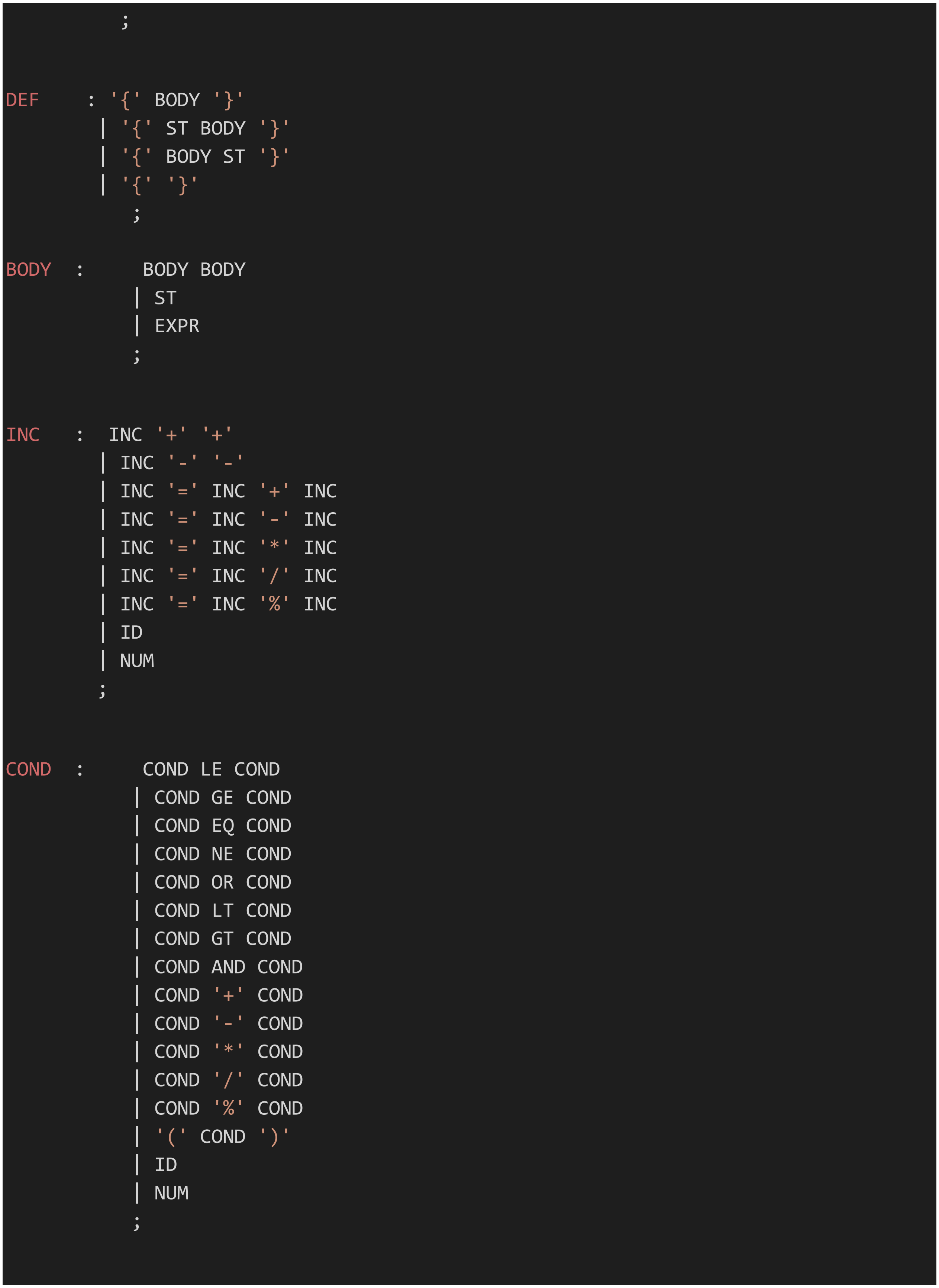
```
%{
#include"y.tab.h"
extern int yylval;
%}

%%

[\t \n] {}
for             return FOR;
[0-9]+     return NUM;
[A-Za-z_]+ return ID;
[char|short|int|long|float|double|bool|void|wchar_t|signed|unsigned] return KEY;
"<="         return LE;
">="         return GE;
"=="         return EQ;
"!="          return NE;
"||"            return OR;
"&&"         return AND;
">"           return GT;
"<"           return LT;
"//code"          return STMT;
.                return yytext[0];
"$\n"           return END;

%%

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

YACC Code:

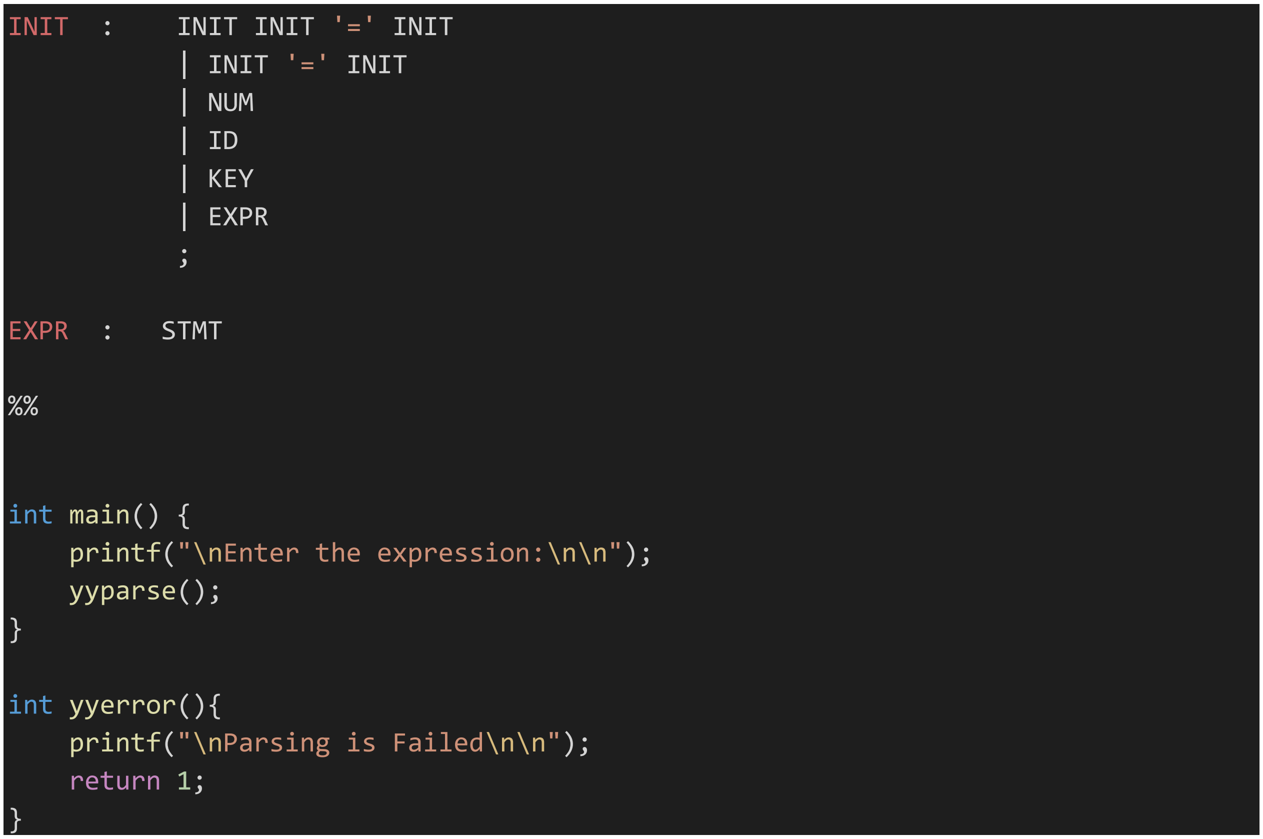
```
%{
#include <stdio.h>
#include <stdlib.h>
int yylex();
int yyparse();
%}

%token END
%token KEY ID NUM FOR LE GE EQ NE OR AND GT LT STMT
%right '='
%left AND OR
```



```
%left '<' '>' LE GE EQ NE
%left '+''-'
%left '*''/'
%left '!'

%%

S         : STS END{printf("\nParsing is Successful\n\n"); exit(0);}

STS       : STS ST
          | ST
          ;

ST        : FOR '(' INIT ';' COND ';' INC ')' DEF
          | FOR '(' INIT ';' COND ';' INC ')' EXPR
          | FOR '(' INIT ';' COND ';' INC ')' ';'

          | FOR '(' INIT ';' COND ';' ')' DEF
          | FOR '(' INIT ';' COND ';' ')' EXPR
          | FOR '(' INIT ';' COND ';' ')' ';'

          | FOR '(' INIT ';' ';' INC ')' DEF
          | FOR '(' INIT ';' ';' INC ')' EXPR
          | FOR '(' INIT ';' ';' INC ')' ';'

          | FOR '(' ';' COND ';' INC ')' DEF
          | FOR '(' ';' COND ';' INC ')' EXPR
          | FOR '(' ';' COND ';' INC ')' ';'

          | FOR '(' INIT ';' ';' ')' DEF
          | FOR '(' INIT ';' ';' ')' EXPR
          | FOR '(' INIT ';' ';' ')' ';'

          | FOR '(' ';' COND ';' ')' DEF
          | FOR '(' ';' COND ';' ')' EXPR
          | FOR '(' ';' COND ';' ')' ';'

          | FOR '(' ';' ';' INC ')' DEF
          | FOR '(' ';' ';' INC ')' EXPR
          | FOR '(' ';' ';' INC ')' ';'

          | FOR '('  ';'  ';'  ')' DEF
          | FOR '('  ';'  ';'  ')' EXPR
          | FOR '('  ';'  ';'  ')' ';'
```

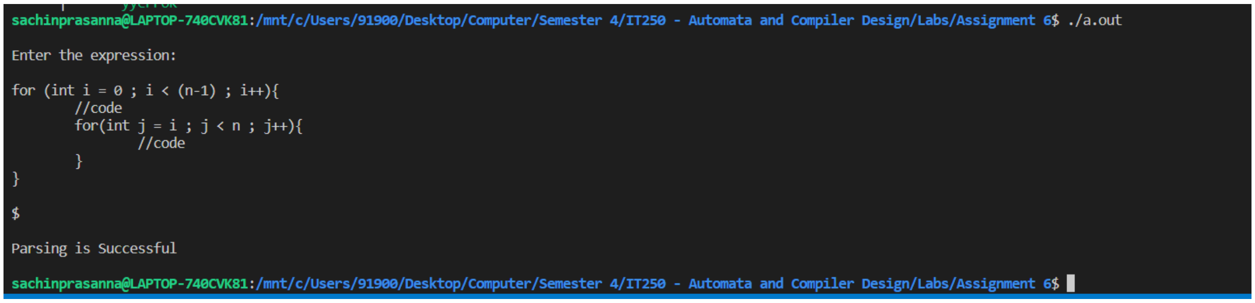


```
          ;


DEF    : '{' BODY '}'
        | '{' ST BODY '}'
        | '{' BODY ST '}'
        | '{' '}'
           ;

BODY  :     BODY BODY
           | ST
           | EXPR
           ;


INC   :  INC '+' '+'
        | INC '-' '-'
        | INC '=' INC '+' INC
        | INC '=' INC '-' INC
        | INC '=' INC '*' INC
        | INC '=' INC '/' INC
        | INC '=' INC '%' INC
        | ID
        | NUM
        ;


COND  :     COND LE COND
           | COND GE COND
           | COND EQ COND
           | COND NE COND
           | COND OR COND
           | COND LT COND
           | COND GT COND
           | COND AND COND
           | COND '+' COND
           | COND '-' COND
           | COND '*' COND
           | COND '/' COND
           | COND '%' COND
           | '(' COND ')'
           | ID
           | NUM
           ;
```

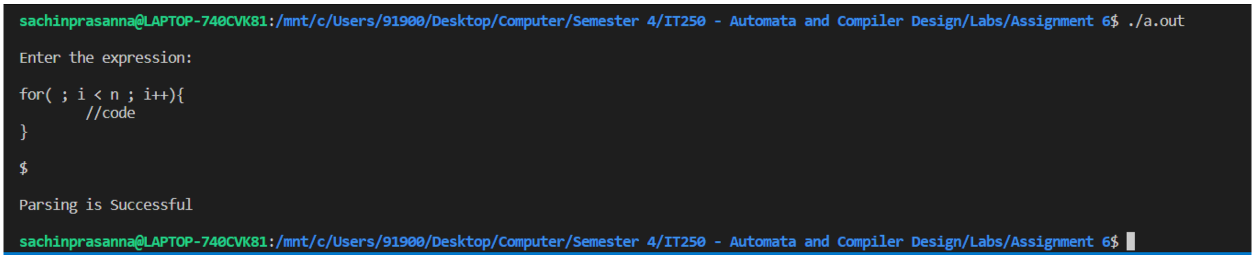


```
INIT   :    INIT INIT '=' INIT
            | INIT '=' INIT
            | NUM
            | ID
            | KEY
            | EXPR
            ;

EXPR   :   STMT

%%


int main() {
    printf("\nEnter the expression:\n\n");
    yyparse();
}

int yyerror(){
    printf("\nParsing is Failed\n\n");
    return 1;
}
```

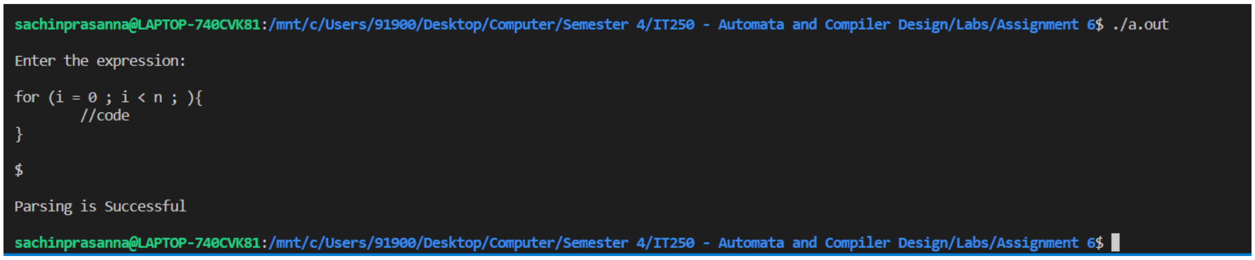
Outputs:



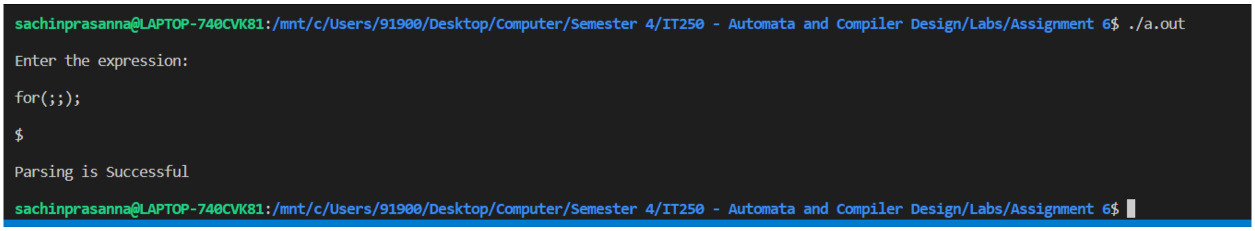
```
sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6$ ./a.out

Enter the expression:

for (int i = 0 ; i < (n-1) ; i++){
        //code
        for(int j = i ; j < n ; j++){
                //code
        }
}

$

Parsing is Successful

sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6$ 
```



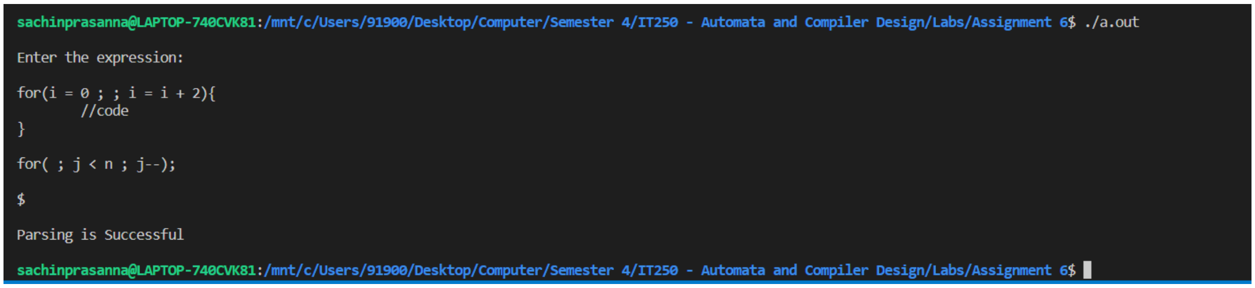
```
sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6$ ./a.out

Enter the expression:

for( ; i < n ; i++){
        //code
}

$

Parsing is Successful

sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6$ 
```



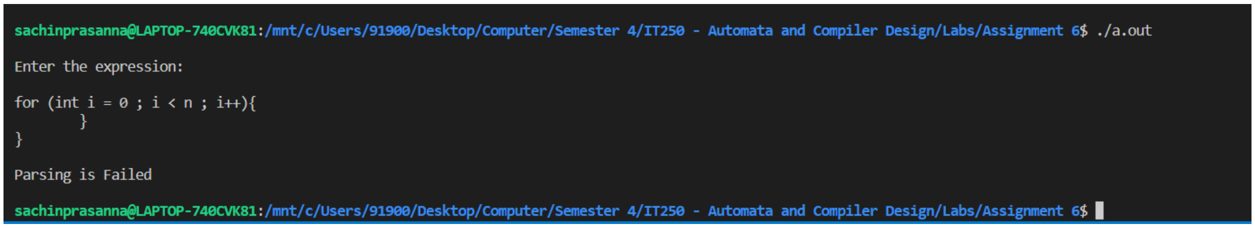
```
sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6$ ./a.out
Enter the expression:
for (i = 0 ; i < n ; ){
        //code
}
$
Parsing is Successful
sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6$
```



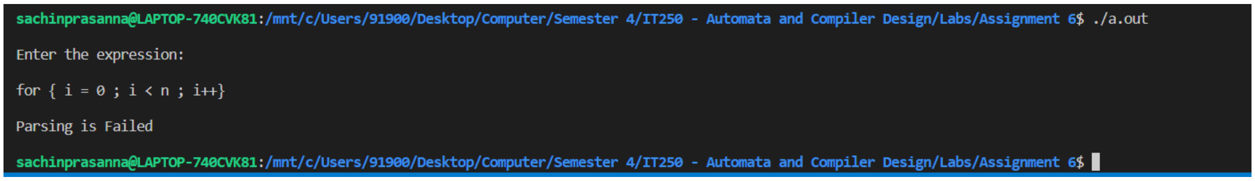
```
sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6$ ./a.out
Enter the expression:
for(;;);
$
Parsing is Successful
sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6$
```



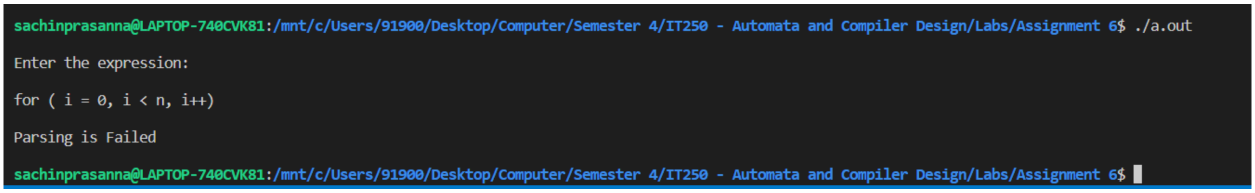
```
sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6$ ./a.out
Enter the expression:
for(i = 0 ; ; i = i + 2){
        //code
}
for( ; j < n ; j--);
$
Parsing is Successful
sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6$
```



```
sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6$ ./a.out
Enter the expression:
for (int i = 0 ; i < n ; i++){
        }
}
Parsing is Failed
sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6$
```



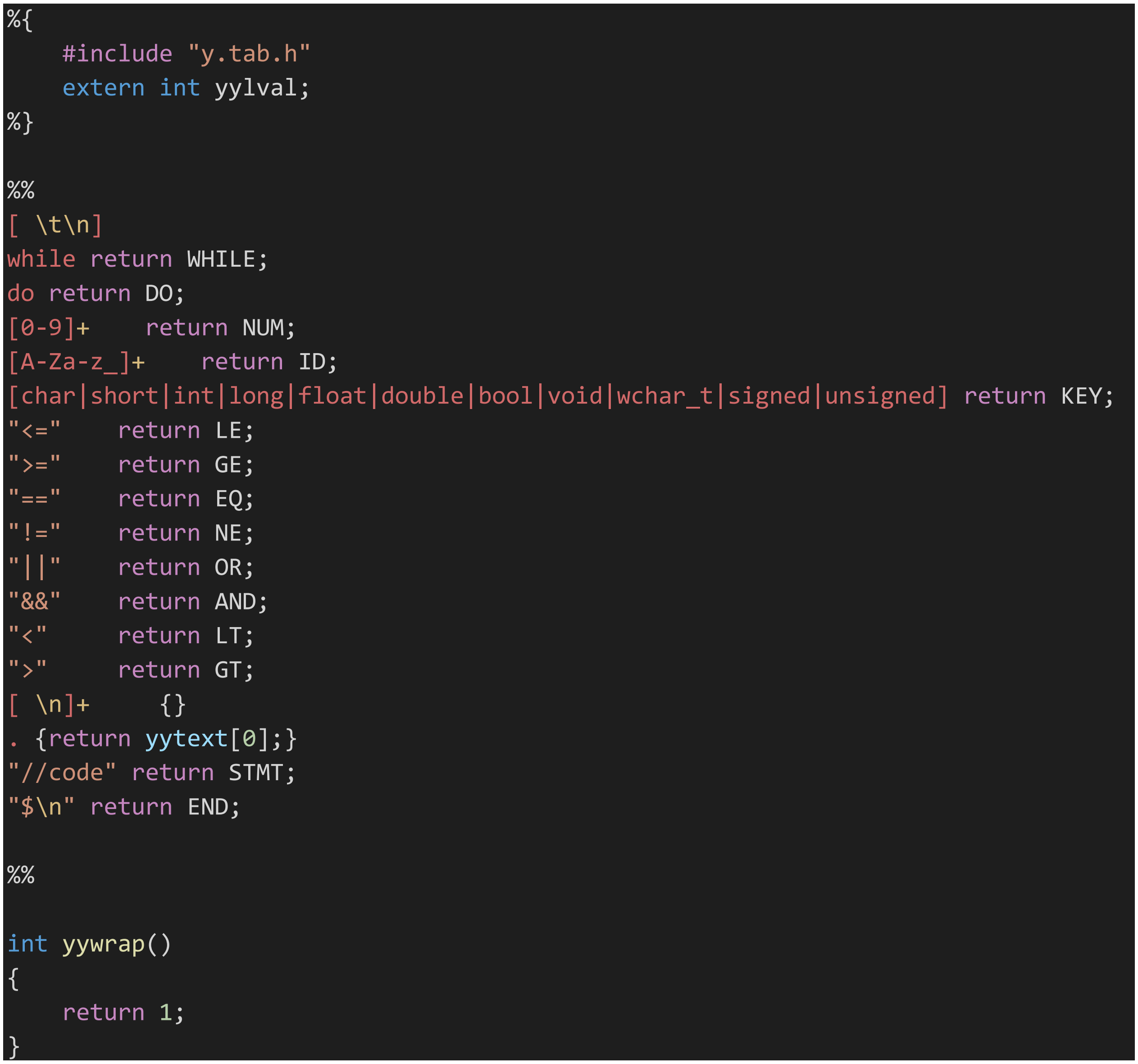
```
sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6$ ./a.out
Enter the expression:
for { i = 0 ; i < n ; i++}
Parsing is Failed
sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6$
```



```
sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6$ ./a.out
Enter the expression:
for ( i = 0, i < n, i++)
Parsing is Failed
sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6$
```

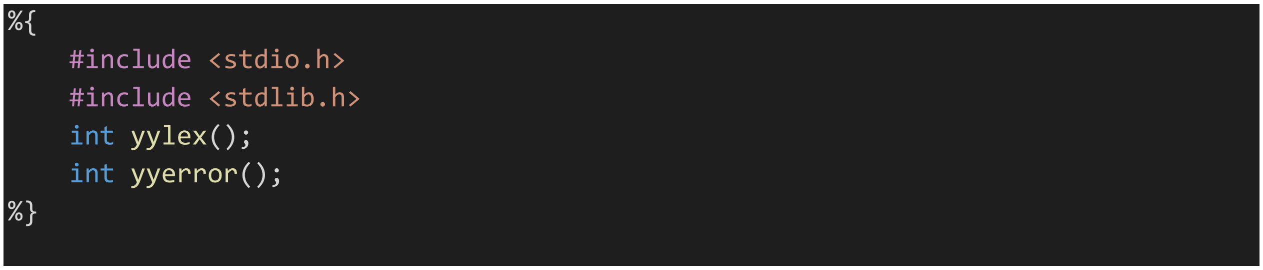
2b)

LEX Code:

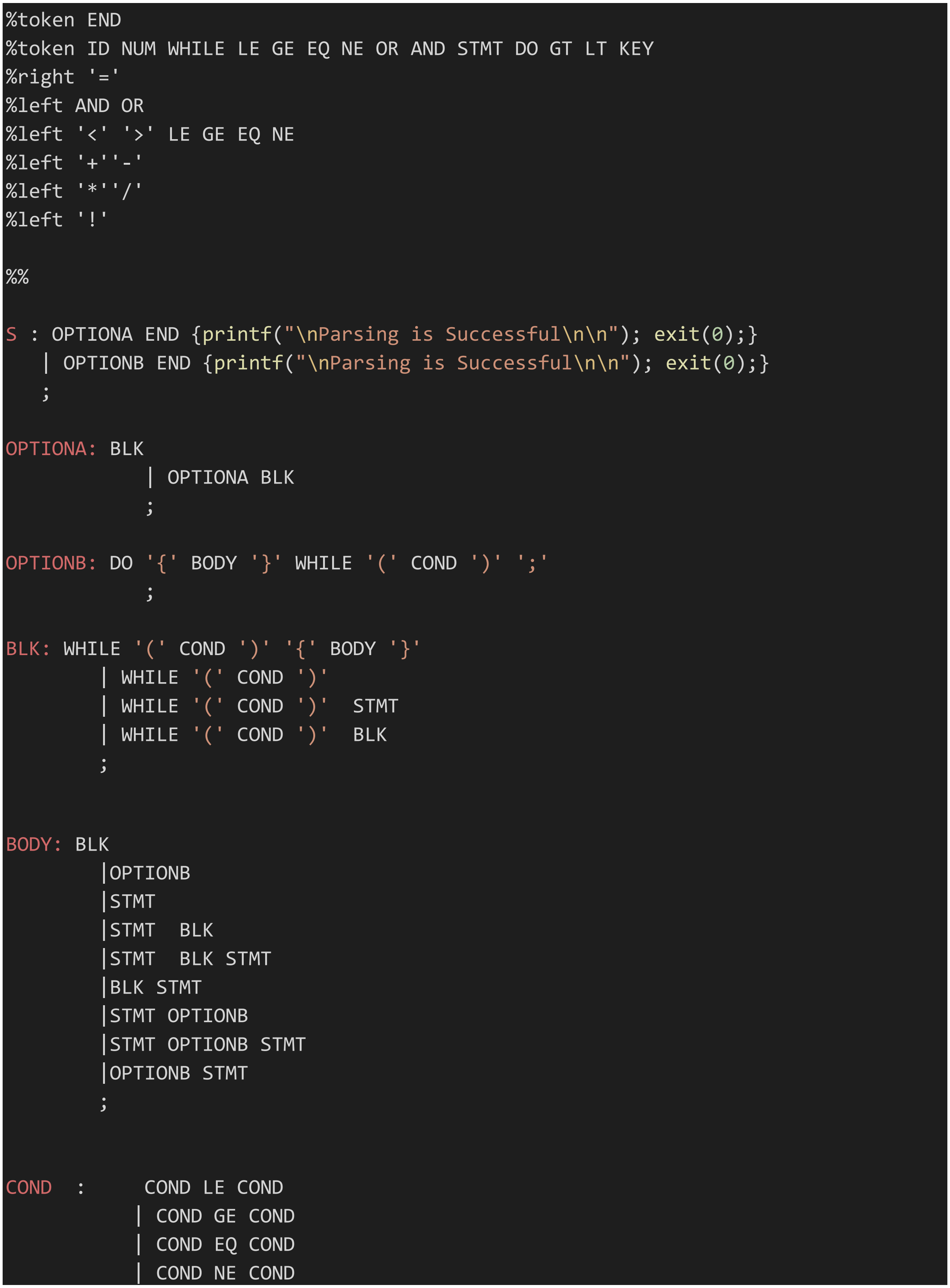


```
%{
    #include "y.tab.h"
    extern int yylval;
%}

%%
[ \t\n]
while return WHILE;
do return DO;
[0-9]+    return NUM;
[A-Za-z_]+    return ID;
[char|short|int|long|float|double|bool|void|wchar_t|signed|unsigned] return KEY;
"<="    return LE;
">="    return GE;
"=="    return EQ;
"!="    return NE;
"||"    return OR;
"&&"    return AND;
"<"     return LT;
">"     return GT;
[ \n]+     {}
. {return yytext[0];}
"//code" return STMT;
"$\n" return END;

%%

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

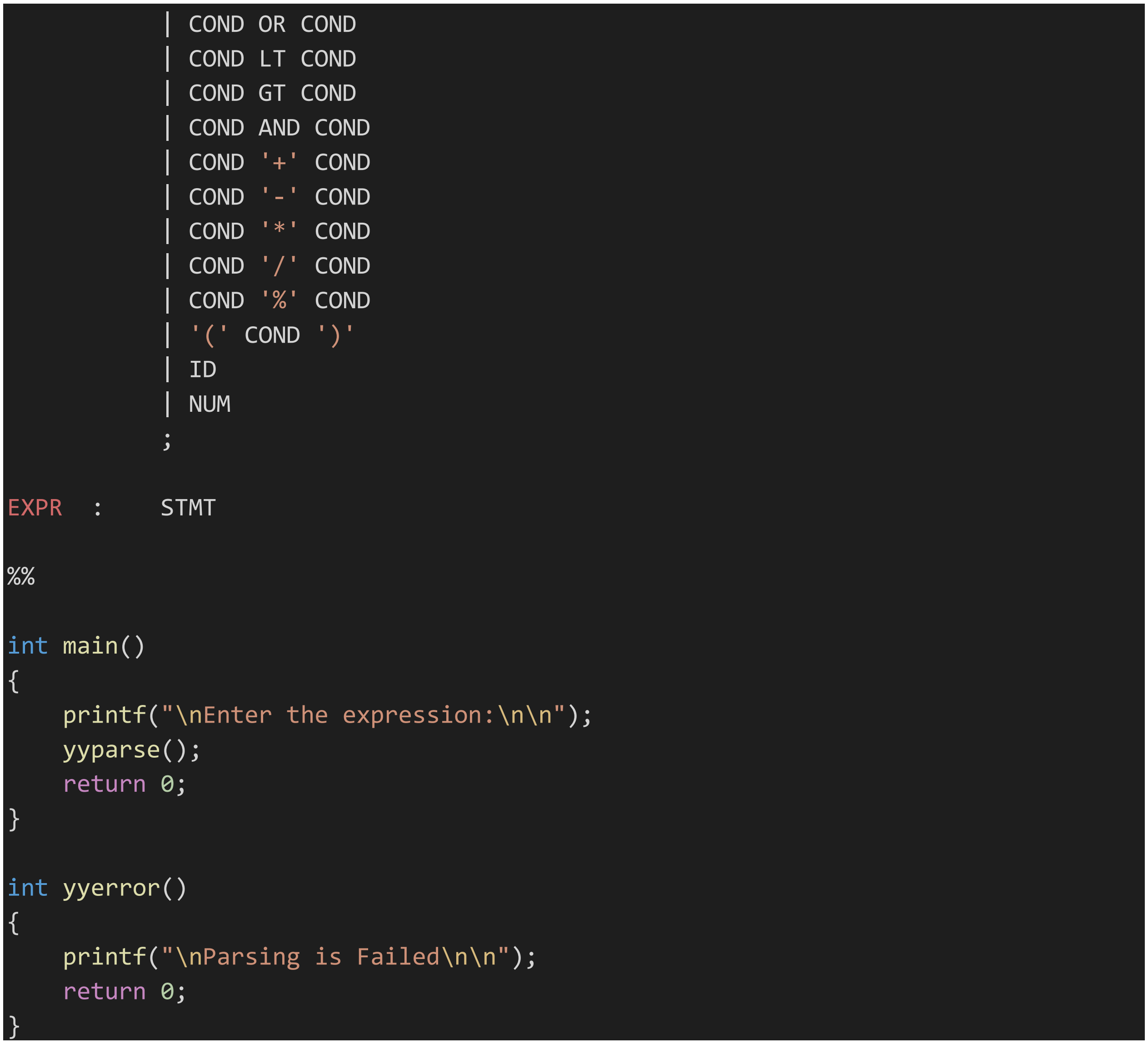
YACC Code:



```
%{
    #include <stdio.h>
    #include <stdlib.h>
    int yylex();
    int yyerror();
%}
```

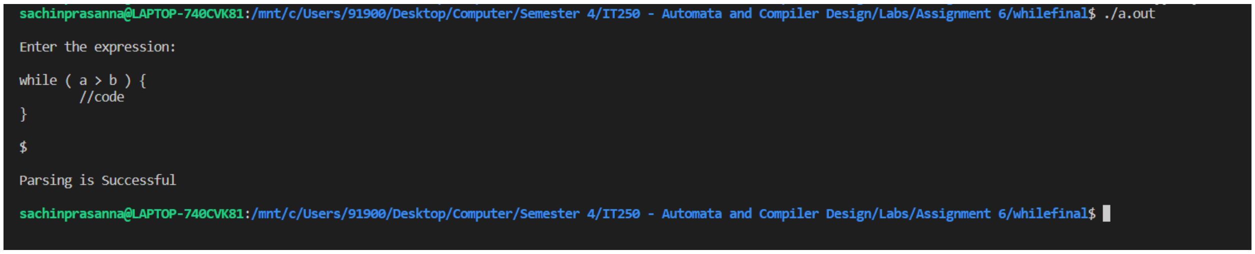


```
%token END
%token ID NUM WHILE LE GE EQ NE OR AND STMT DO GT LT KEY
%right '='
%left AND OR
%left '<' '>' LE GE EQ NE
%left '+''-'
%left '*''/'
%left '!'

%%

S : OPTIONA END {printf("\nParsing is Successful\n\n"); exit(0);}
   | OPTIONB END {printf("\nParsing is Successful\n\n"); exit(0);}
   ;

OPTIONA: BLK
            | OPTIONA BLK
            ;

OPTIONB: DO '{' BODY '}' WHILE '(' COND ')' ';'
            ;

BLK: WHILE '(' COND ')' '{' BODY '}'
        | WHILE '(' COND ')'
        | WHILE '(' COND ')'  STMT
        | WHILE '(' COND ')'  BLK
        ;


BODY: BLK
        |OPTIONB
        |STMT
        |STMT  BLK
        |STMT  BLK STMT
        |BLK STMT
        |STMT OPTIONB
        |STMT OPTIONB STMT
        |OPTIONB STMT
        ;


COND  :     COND LE COND
           | COND GE COND
           | COND EQ COND
           | COND NE COND
```

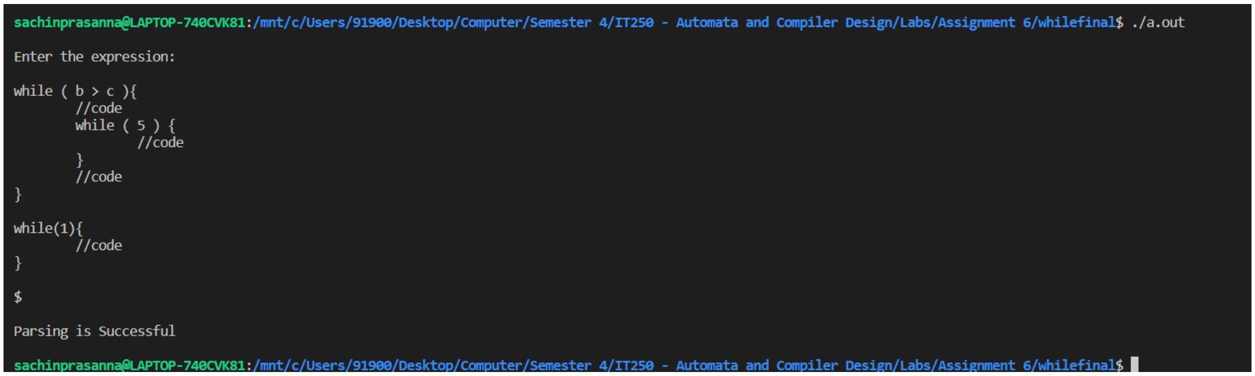


```
           | COND OR COND
           | COND LT COND
           | COND GT COND
           | COND AND COND
           | COND '+' COND
           | COND '-' COND
           | COND '*' COND
           | COND '/' COND
           | COND '%' COND
           | '(' COND ')'
           | ID
           | NUM
           ;

EXPR  :    STMT

%%

int main()
{
    printf("\nEnter the expression:\n\n");
    yyparse();
    return 0;
}

int yyerror()
{
    printf("\nParsing is Failed\n\n");
    return 0;
}
```

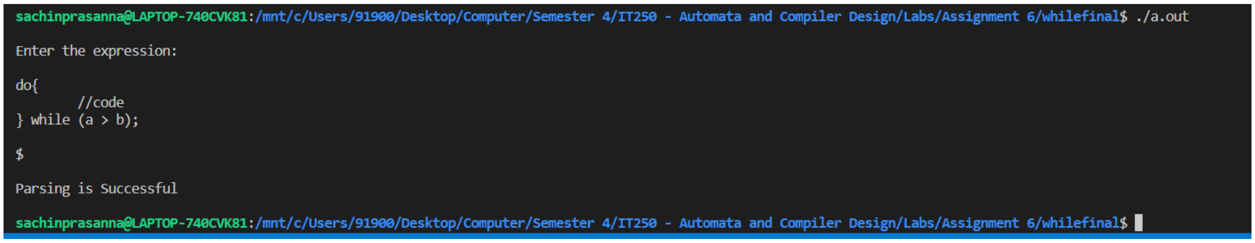
Outputs:



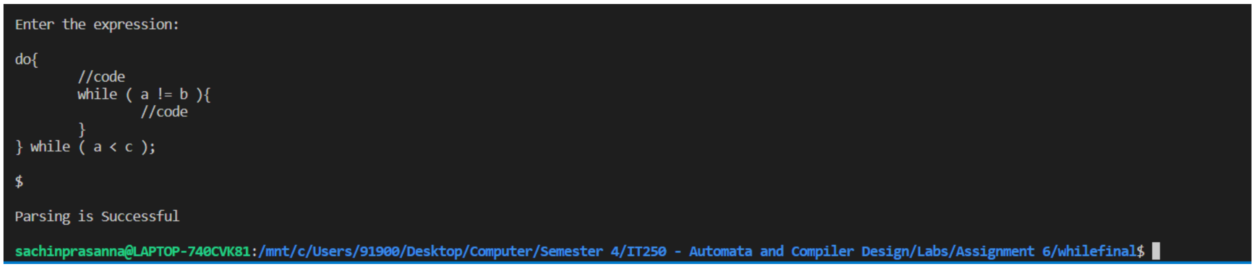
```
sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6/whilefinal$ ./a.out

Enter the expression:

while ( a > b ) {
        //code
}

$

Parsing is Successful

sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6/whilefinal$
```



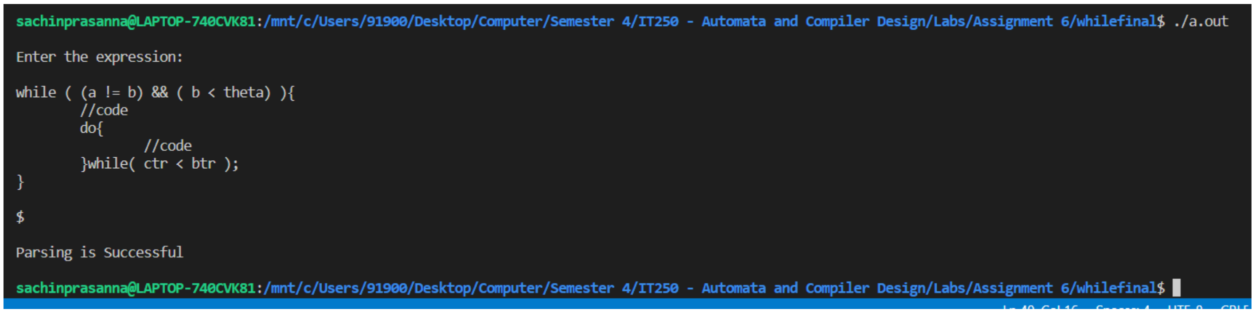
```
sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6/whilefinal$ ./a.out

Enter the expression:

while ( b > c ){
        //code
        while ( 5 ) {
                //code
        }
        //code
}

while(1){
        //code
}

$

Parsing is Successful

sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6/whilefinal$
```



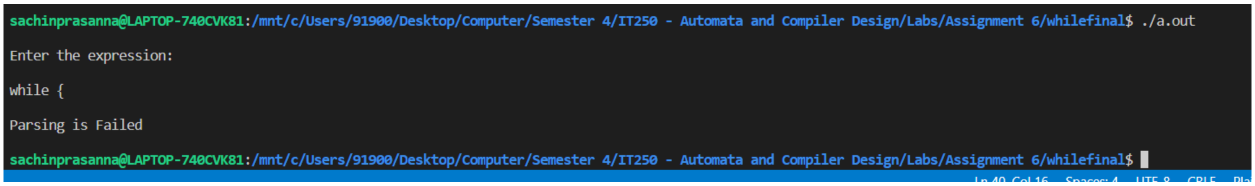
```
sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6/whilefinal$ ./a.out

Enter the expression:

do{
        //code
} while (a > b);

$

Parsing is Successful

sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6/whilefinal$
```



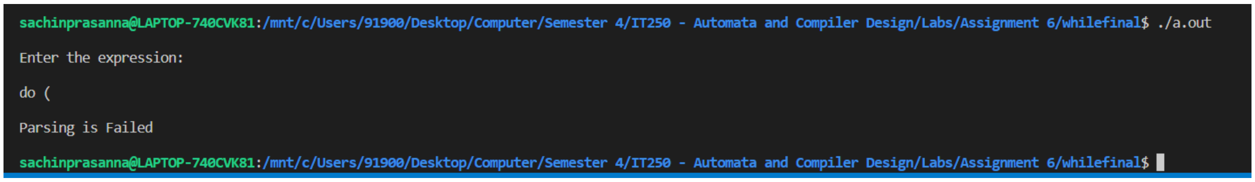
```
Enter the expression:

do{
        //code
        while ( a != b ){
                //code
        }
} while ( a < c );

$

Parsing is Successful

sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6/whilefinal$
```



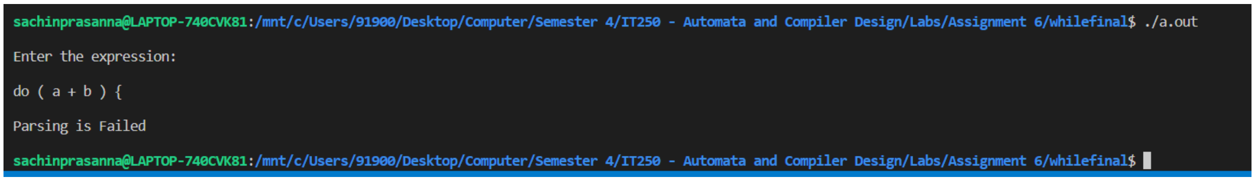
```
sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6/whilefinal$ ./a.out

Enter the expression:

while ( (a != b) && ( b < theta) ){
        //code
        do{
                //code
        }while( ctr < btr );
}

$

Parsing is Successful

sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6/whilefinal$
```



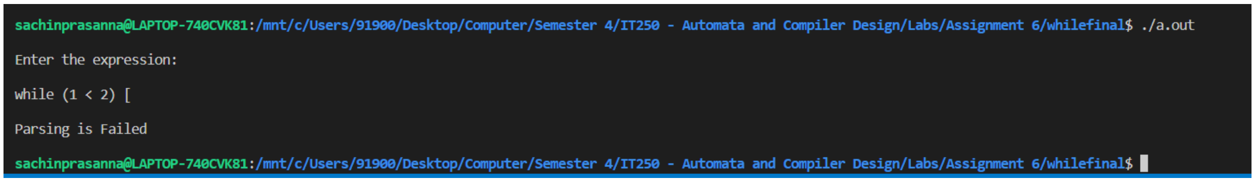
```
sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6/whilefinal$ ./a.out

Enter the expression:

while {

Parsing is Failed

sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6/whilefinal$
```



```
sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6/whilefinal$ ./a.out

Enter the expression:

do (

Parsing is Failed

sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6/whilefinal$
```



```
sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6/whilefinal$ ./a.out
Enter the expression:
do ( a + b ) {
Parsing is Failed
sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6/whilefinal$
```



```
sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6/whilefinal$ ./a.out
Enter the expression:
while (1 < 2) [
Parsing is Failed
sachinprasanna@LAPTOP-740CVK81:/mnt/c/Users/91900/Desktop/Computer/Semester 4/IT250 - Automata and Compiler Design/Labs/Assignment 6/whilefinal$
```

****