**LEX**

%{

    //Definition section

    #include<stdio.h>

    #include<string.h>

    #include<stdlib.h>

    #include "ifelse.tab.h"

%}

st [a-z0-9]\*

%%

if return IF;

else return ELSE;

{st} return STATEMENT;

{st}<{st} return CONDITION;

[(,),{,},;] return yytext[0];

[\t] ;

[\n] ;

. yyerror();

%%

int yywrap(){

    return 1;

}

**YACC**

%{

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

int yylex();

extern char\* yytext;

int yyerror();

int flag = 0;

extern FILE\* yyin;

%}

%token IF ELSE STATEMENT CONDITION

%%

E: S E

| S {

    if(flag==0)

        printf("\nEntered expression is valid\n");

};

S: B {printf("Reduced B to S\n");}

| B ELSE'{'S'}' {printf("Reduced B ELSE'{'S'}' to S\n");}

| A {printf("Reduced A to S\n");}

;

B: IF'('CONDITION')''{'S'}' {printf("Reduced IF'('CONDITION')''{'S'}' to B\n");}

;

A: STATEMENT';' {printf("Reduced STATEMENT to A\n");}

;

%%

//driver code

void main()

{

    FILE\* f1 = fopen("try.txt", "r");

    if(f1==NULL){

        printf("Error while openeing file\n");

        exit(0);

    }

    yyin = f1;

    yyparse();

    if(flag==0)

    printf("\nEntered syntax is Valid\n\n");

}

int yyerror()

{

printf("\nEntered syntax is Invalid %s\n\n", yytext);

flag = 1;

return 0;

}