**Exper4.l:**

%{

#include<stdio.h>

#include "y.tab.h"

%}

L [A-Za-z]

D [0-9]

id {L}({L}|{D})\*

%%

"if" {return IF;}

"else" {return ELSE;}

"for" {return FOR;}

"do" {return DO;}

"while" {return WHILE;}

"++" {return INC;}

"--" {return DEC;}

"||" {return OR;}

"&&" {return AND;}

"!" {return NOT;}

"switch" {return SWITCH;}

"case" {return CASE;}

"break" {return BREAK;}

"default" {return DEFAULT;}

[0-9]+(\.[0-9]+)? {return NUM;}

{id} {return id;}

"<"|"<="|">"|">="|"=="|"!=" {return relop;}

[-/;=+\*,\(\)\{\}:] {return yytext[0];}

[ ] {}

\n {}

%%

int yywrap()

{

return 1;

}

**Exper4.y**

%{

#include <stdio.h>

%}

%token id NUM OR AND NOT relop TRUE FALSE INC DEC IF ELSE DO WHILE uminus FOR SWITCH CASE BREAK DEFAULT

%right '='

%left '+' '-'

%left '\*' '/'

%right '^'

%nonassoc uminus

%left OR

%left AND

%nonassoc NOT

%%

S1 : S1 S

| S

;

S : AS ';' {printf("Assignment statement accepted \n");}

|IFS {printf("If statement is accepted \n");}

|IFES {printf("If else statement is accepted\n");}

|WS {printf("While statement is accepted\n");}

|DWS {printf("Do while statement is accepted\n");}

|FORS {printf("For statement is accepted\n");}

|SS {printf("Switch statement is accepted");}

;

SS : SWITCH'('E')''{' CV '}'

;

CV : CASE E ':' S1 BREAK ';'

| CASE E ':' S1 BREAK ';' CV

| CASE E ':' S1 BREAK ';' DEFAULT ':' S1

;

AS : id '=' E

;

E : E'+'E

|E'-'E

|E'\*'E

|E'/'E

|E'^'E

|'-' E %prec uminus

|id

|NUM

;

IFS: IF'('BE')''{'S1'}'

;

BE : BE OR BE

| BE AND BE

| NOT BE

|id relop id

|TRUE

|FALSE

;

IFES : IF'('BE')''{'S1'}'ELSE'{'S1'}'

;

WS : WHILE '('BE')''{'S1'}'

;

DWS : DO '{'S1'}'WHILE'('BE')'';'

;

FORS : FOR'('IS';'BE';'MS')''{'S1'}'

;

IS : AS

| IS ',' AS

;

MS : IS

| id INC

| INC id

| id DEC

| DEC id

;

%%

void main()

{

yyparse();

}

int yyerror(char \*msg)

{

printf("%s\n",msg);

}

