National Institute Of Technology Surathkal Mangalore Karnataka-575025

Department Of Information Technology

Lab Assignment :- 06

Name:- Chikkeri Chinmaya

Roll Number:- 211IT017

Branch:- Information Technology (B.Tech)

Section :- S13

Course:- Automata And Compiler Design (IT251)

Submitted To:-

Anupama H C Mam

1. a) “ if – else ” programming construct of "C" programming language

ifelse.l

```
%{

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

alpha [A-Za-z]
digit [0-9]

%%

[\t \n]
if          return IF;
else        return ELSE;
elseif      return ELSEIF;
{digit}+    return NUM;
{alpha}({alpha}|{digit})* return ID;
"<="        return LE;
">="        return GE;
"=="        return EQ;
"!="        return NE;
"||"            return OR;
"&&"        return AND;
.               return yytext[0];

%%
```

Ifelse.y

```
%{
#include <stdio.h>
#include <stdlib.h>
%}

%token ID NUM IF ELSE ELSEIF LE GE EQ NE OR AND
%right "="
%left OR AND
%left '>' '<' LE GE EQ NE
%left '+' '-'
%left '*' '/'
```

```
%right UMINUS
%left '!'

%%

S       : ST {printf("Parsing is Successful\n"); exit(0);}
ST      : IF '(' E ')' DEF LADDER ELSE DEF
        ;

LADDER : LADDER LADDER
        | ELSEIF '(' E ')' DEF
        |
        ;


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

E       : ID '=' E
        | E '+' E
        | E '-' E
        | E '*' E
        | E '/' E
        | E '<' E
        | E '>' E
        | E LE E
        | E GE E
        | E EQ E
        | E NE E
        | E OR E
        | E AND E
        | E '+' '+'
        | E '-' '-'
        | ID
        | NUM
        ;


E2  : E'<'E
        | E'>'E
        | E LE E
```

```
        | E GE E
        | E EQ E
        | E NE E
        | E OR E
        | E AND E
        ;
%%


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

int yyerror(char* s) {
printf("Parsing is Failed\n");
}
```

b) “ switch case ” statements of "C" programming language.

Switch.l

```
%{

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

alpha [A-Za-z]
digit [0-9]

%%

[\t \n]
switch      return SWITCH;
case        return CASE;
default     return DEFAULT;
break       return BREAK;
{digit}+    return NUM;
{alpha}({alpha}|{digit})* return ID;
"<="        return LE;
">="        return GE;
"=="        return EQ;
"!="        return NE;
"||"            return OR;
"&&"        return AND;
```

```
.               return yytext[0];

%%
```

Switch.y

```
%{
#include <stdio.h>
#include <stdlib.h>
%}

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

%%

S       : ST {printf("Parsing is Successful\n"); exit(0);}
ST      : SWITCH '(' ID ')' '{' CASEBLOCK '}'
        ;

CASEBLOCK : CASEBLOCK CASEBLOCK
          | CASE NUM ':' BODY BREAK ';'
          | CASE NUM ':' BODY BREAK ';' DEFAULTBLOCK
          ;

DEFAULTBLOCK : DEFAULT ':' BODY
             |
             ;



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


BODY  : BODY BODY
        | E ';'
        | ST
        |
        ;
```

```
E       : ID '=' E
        | E '+' E
        | E '-' E
        | E '*' E
        | E '/' E
        | E '<' E
        | E '>' E
        | E LE E
        | E GE E
        | E EQ E
        | E NE E
        | E OR E
        | E AND E
        | E '+' '+'
        | E '-' '-'
        | ID
        | NUM
        ;


E2  : E'<'E
        | E'>'E
        | E LE E
        | E GE E
        | E EQ E
        | E NE E
        | E OR E
        | E AND E
        ;
%%


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

int yyerror(char* s) {
printf("Parsing is Failed\n");
}
```

2)

a)

b) “ while ’’ and “ do while ” loop Construc

```
%{

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

alpha [A-Za-z]
digit [0-9]

%%

[\t \n]
for             return FOR;
{digit}+    return NUM;
{alpha}({alpha}|{digit})* return ID;
"<="        return LE;
">="        return GE;
"=="        return EQ;
"!="        return NE;
"||"            return OR;
"&&"        return AND;
.               return yytext[0];

%%
```

Do while

```
%{

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

alpha [A-Za-z]
digit [0-9]
%%
[ \t\n]
while    return WHILE;
{digit}+    return NUM;
{alpha}({alpha}|{digit})*    return ID;
"<="    return LE;
">="    return GE;
"=="    return EQ;
"!="    return NE;
"||"    return OR;
"&&"    return AND;
.    return yytext[0];
```

```
%%

A1.y
%{
#include <stdio.h>
#include <stdlib.h>
%}
%token ID NUM WHILE LE GE EQ NE OR AND
%right '='
%left AND OR
%left '<' '>' LE GE EQ NE
%left '+''-'
%left '*''/'
%right UMINUS
%left '!'
%%
S        : ST1 {printf("Parser is accepted.\n");exit(0);};
ST1    :    WHILE'(' E2 ')' '{' ST '}'
ST      :     ST ST
          | E';'
          ;
E       : ID'='E
          | E'+'E
          | E'-'E
          | E'*'E
          | E'/'E
          | E'<'E
          | E'>'E
          | E LE E
          | E GE E
          | E EQ E
          | E NE E
          | E OR E
          | E AND E
          | ID
          | NUM
          ;
E2     : E'<'E
          | E'>'E
          | E LE E
          | E GE E
          | E EQ E
          | E NE E
          | E OR E
          | E AND E
          | ID
          | NUM
          ;
```

```
%%


main()
{
   printf("Enter the exp: ");
   yyparse();
}
int yyerror(char* s) {
printf("Parser is not accepted\n");
}
```