

## Exp 7f : YACC – WHILE

### LEX

1. Start
2. %% - rule section

```
"while" return WHILE
[ \t\n]
[0-9]+ return NUM
[a-zA-Z][a-zA-Z0-9]* return ID
\"[^\"]*" return STRING
"<" return L
">" return G
"<=" return LE
">=" return GE
"==" return EE
"!=" return NE
"++" return INC
"--" return DEC
"||" return OR
"&&" return AND
. return yytext[0]
```
3. yywrap() return 1
4. Stop

### YACC

1. Start
2. %token WHILE L G LE GE EE NE INC DEC OR AND ID NUM STRING
3. %% rule section

```
while : WHILE '(' cond ')' '{' stmt '}' { print "valid while loop" } ;
cond : scond | scond AND cond | scond OR cond ;
scond : nid | nid relop nid ;
nid : ID | NUM ;
relop : L | G | LE | GE | EE | NE ;
stmt : ID '(' STRING other ')' ';' stmt | E ';' stmt | ;
other : ';' ID other | ';' '&' ID other | ;
E : ID '=' E | E '+' E | E '-' E | E '*' E | E '/' E | E INC | E DEC | nid | '(' nid ')';
```
4. yyerror() to handle error
5. in main() call yyparse()

### while – Lex

```
%{
#include<stdio.h>
#include "y.tab.h"
}%

%%

"while" { return WHILE; }

[ \t\n]
[0-9]+ { return NUM; }
[a-zA-Z][a-zA-Z0-9]* { return ID; }
\"[^\"]*" { return STRING; }
```

```

"<" { return L; }
">" { return G; }
"<=" { return LE; }
">=" { return GE; }
"==" { return EE; }
"!=" { return NE; }
"++" { return INC; }
"--" { return DEC; }
"||" { return OR; }
"&&" { return AND; }
. { return yytext[0]; }
%%

```

```

int yywrap(){
    return 1;
}

```

### while – YACC

```

%{
    #include<stdio.h>
}%

```

```

%token WHILE L G LE GE EE NE INC DEC OR AND ID NUM STRING

```

```

%%
while : WHILE '(' cond ')' '{' stmt '}' { printf("valid while loop\n"); };
cond : scond | scond AND cond | scond OR cond ;
scond : nid | nid relop nid ;
nid : ID | NUM ;
relop : L | G | LE | GE | EE | NE ;
stmt : ID '(' STRING other ')' ';' stmt | E ';' stmt | ;
other : ',' ID other | ',' '&' ID other | ;

```

```

E : ID '=' E
| E '+' E
| E '-' E
| E '*' E
| E '/' E
| E INC
| E DEC
| nid
| '(' nid ')'
;
%%

```

```

int yyerror(){
    printf("invalid while loop\n");
    return 1;
}

```

```

int main(){
    printf("Enter while loop (press ctrl+D to get output)\n");
    yyparse();
    return 0;
}

```

## output

```

• deadpool@daredevil:~/Desktop/s7-CD/03 YACC/Loops & Statements/WHILE$ flex while.l
• deadpool@daredevil:~/Desktop/s7-CD/03 YACC/Loops & Statements/WHILE$ yacc -d while.y
while.y: warning: 30 shift/reduce conflicts [-Wconflicts-sr]
while.y: note: rerun with option '-Wcounterexamples' to generate conflict counterexamples
• deadpool@daredevil:~/Desktop/s7-CD/03 YACC/Loops & Statements/WHILE$ gcc lex.yy.c y.tab.c -o while
y.tab.c: In function 'yyparse':
y.tab.c:1101:16: warning: implicit declaration of function 'yylex' [-Wimplicit-function-declaration]
 1101 |         yychar = yylex ();
      |                  ^~~~~~
y.tab.c:1242:7: warning: implicit declaration of function 'yyerror'; did you mean 'yyerrok'? [-Wimplicit-function-declaration]
 1242 |         yyerror (YY_("syntax error"));
      |         ^~~~~~
      |         yyerrok
• deadpool@daredevil:~/Desktop/s7-CD/03 YACC/Loops & Statements/WHILE$ ./while
Enter the while loop (press ctrl+D to get output)
while ( i < n ){
    a = b + c - d * e / f ;
    b++;
    c--;
    printf ( " %d %d %d\n " ,d,e,f );
}
valid while loop
• deadpool@daredevil:~/Desktop/s7-CD/03 YACC/Loops & Statements/WHILE$ ./while
Enter the while loop (press ctrl+D to get output)
while ( i < n ){
    a = b + c - d * e / f ;
    b++;
    c--;
    printf ( " %d %d %d\n " ,d,e,f )
}
invalid while loop
• deadpool@daredevil:~/Desktop/s7-CD/03 YACC/Loops & Statements/WHILE$ █

```