Exp 7h : YACC – SWITCH

LEX

1. Start
2. %% - rule section
   "switch" return SWITCH
   "case" return CASE
   "break" return BREAK
   "default" return DEFAULT
   [\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() retturn 1
4. Stop

YACC

1. Start
2. %token SWITCH CASE DEFAULT BREAK L G LE GE EE NE INC DEC OR AND ID NUM STRING
3. %% rule section
   switch : SWITCH '(' nid ')' '{' casedefault '}' { print "valid switch case” } ;
   casedefault : case | case default ;
   case : CASE nid ':' stmt BREAK ';' case | ;
   default : DEFAULT ':' stmt BREAK ';' ;
   nid : ID | NUM ;
   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()

switch – Lex

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

%%
"switch" { return SWITCH; }
"case" { return CASE; }
```

```
"break" { return BREAK; }
"default" { return DEFAULT; }

[ \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;
}
```

switch – YACC

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

%token SWITCH CASE DEFAULT BREAK L G LE GE EE NE INC DEC OR AND ID NUM
STRING

%%
switch : SWITCH '(' nid ')' '{' casedefault '}' { printf("valid switch case\n"); return 0; } ;
casedefault : case | case default ;
case : CASE nid ':' stmt BREAK ';' case | ;
default : DEFAULT ':' stmt BREAK ';' ;

nid : ID | NUM ;
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 switch case\n");
   return 1;
}

int main(){
   printf("Enter the switch case (press ctrl+D to get output)\n");
   yyparse();
   return 0;
}
```

output

```
deadpool@daredevil:~/Desktop/s7-CD/03 YACC/Loops & Statements/SWITCH$ flex switch.l
deadpool@daredevil:~/Desktop/s7-CD/03 YACC/Loops & Statements/SWITCH$ yacc -d switch.y
switch.y: warning: 30 shift/reduce conflicts [-Wconflicts-sr]
switch.y: note: rerun with option '-Wcounterexamples' to generate conflict counterexamples
deadpool@daredevil:~/Desktop/s7-CD/03 YACC/Loops & Statements/SWITCH$ gcc lex.yy.c y.tab.c -o switch
y.tab.c: In function ‘yyparse’:
y.tab.c:1111:16: warning: implicit declaration of function ‘yylex’ [-Wimplicit-function-declaration]
 1111 |       yychar = yylex ();
      |                ^~~~~
y.tab.c:1252:7: warning: implicit declaration of function ‘yyerror’; did you mean ‘yyerrok’? [-Wimplicit-function-declaration]
 1252 |       yyerror (YY_("syntax error"));
      |       ^~~~~~~
      |       yyerrok
deadpool@daredevil:~/Desktop/s7-CD/03 YACC/Loops & Statements/SWITCH$ ./switch
Enter the switch case
 : switch(a){
case 1 : i++;
        break;
case 2 : printf("case 2\n");
        break;
default : j--;
        break;
}
valid switch case
deadpool@daredevil:~/Desktop/s7-CD/03 YACC/Loops & Statements/SWITCH$ ./switch
Enter the switch case
 : switch(a){
case 1 :  id = b + c;
        break;
case 2 : printf( " %d\n ",id );
        break;
}
valid switch case
deadpool@daredevil:~/Desktop/s7-CD/03 YACC/Loops & Statements/SWITCH$ ./switch
Enter the switch case
 : switch(a){
case 1 : i++;
        break;
        j--;
invalid switch case
```