

```
1: // $Id: extern.h,v 1.1 2018-04-02 15:31:00-07 - - $
2:
3: #ifndef __EXTERN_H__
4: #define __EXTERN_H__
5:
6: int yylex();
7: int yyparse();
8:
9: void badchar (char);
10: void yyerror (const char*);
11:
12: extern char* yytext;
13:
14: #define YYSTYPE double
15: #include "yyparse.h"
16:
17: #endif
18:
```

```
1: /* $Id: lexer.l,v 1.1 2018-04-02 15:31:00-07 - - $ */
2:
3: %{
4:
5: #include "extern.h"
6:
7: %}
8:
9: %option noyywrap
10:
11: MANTISSA      ([:digit:]+\.[[:digit:]]*|\.[[:digit:]]+)
12: EXPONENT      ([Ee][+-]?[:digit:]+)
13: INFINITY      ([Ii][Nn][Ff][[:isalnum:]]*)
14: NAN           ([Nn][Aa][Nn][[:isalnum:]]*)
15: NUMBER        ({MANTISSA})({EXPONENT})?|INFINITY|NAN
16:
17:
18: %%
19:
20: "#" . *
21: [ \t] +
22: \n          { return '\n'; }
23: {NUMBER}    { return NUMBER; }
24: "+"         { return '+'; }
25: "-"         { return '-'; }
26: "*"         { return '*'; }
27: "/"         { return '/'; }
28:
29: %%
```

```
1: // $Id: parser.y,v 1.1 2018-04-02 15:31:00-07 - - $
2:
3: %{
4:
5: #include <stdio.h>
6: #include <stdlib.h>
7:
8: #include "extern.h"
9:
10: %}
11:
12: %token NUMBER
13:
14: %%
15:
16: input : input expr '\n' { printf (".15g\n", $2); }
17:      | '\n'           { printf ("\n"); }
18:      | error '\n'      { printf ("error\n"); }
19:      |
20:      ;
21:
22: expr  : NUMBER          { $$ = atof (yytext); }
23:      | expr expr '+'    { $$ = $1 + $2; }
24:      | expr expr '-'    { $$ = $1 - $2; }
25:      | expr expr '*'    { $$ = $1 * $2; }
26:      | expr expr '/'    { $$ = $1 / $2; }
27:      ;
28:
29: %%
30:
```

```
1: // $Id: main.cpp,v 1.1 2018-04-02 15:31:00-07 - - $
2:
3: #include <stdio.h>
4: #include <stdlib.h>
5:
6: #include "extern.h"
7:
8: int status = EXIT_SUCCESS;
9:
10: void yyerror (const char* error) {
11:     printf ("%s\n", error);
12:     status = EXIT_FAILURE;
13: }
14:
15: int main () {
16:     yyparse();
17:     return status;
18: }
```

```
1: # $Id: Makefile,v 1.1 2018-04-02 15:31:00-07 - - $
2:
3: CPP      = g++ -g -O0 -std=gnu++17 -Wall -Wextra -Wold-style-cast
4: CPPYY    = g++ -g -O0 -std=gnu++17 -Wno-register
5: OBJJS    = main.o yyparse.o yylex.o
6: GENS     = yyparse.h yyparse.cpp yylex.cpp
7: SRCS     = extern.h lexer.l parser.y main.cpp Makefile
8:
9: all : rpn
10:
11: rpn : ${OBJJS}
12:      ${CPP} -o rpn ${OBJJS}
13:
14: %.o : %.cpp
15:      checksource $<
16:      cpplint.py.perl $<
17:      ${CPP} -c $<
18:
19: yyparse.cpp yyparse.h : parser.y
20:      bison --defines=yyparse.h --output=yyparse.cpp parser.y
21:
22: yyparse.o : yyparse.cpp
23:      ${CPPYY} -c yyparse.cpp
24:
25: yylex.cpp : yyparse.h lexer.l
26:      flex --outfile=yylex.cpp lexer.l
27:
28: yylex.o : yylex.cpp
29:      ${CPPYY} -Wno-register -c yylex.cpp
30:
31: spotless : clean
32:      - rm rpn Makefile.deps Listing.{ps,pdf}
33:
34: clean :
35:      - rm ${OBJJS} ${GENS}
36:
37: ci :
38:      - cid + ${SRCS}
39:
40: lis : ${SRCS}
41:      mkpspdf Listing.ps ${SRCS}
42:
43: Makefile.deps : yyparse.cpp yylex.cpp
44:      ${CPP} -MM yyparse.cpp yylex.cpp main.cpp
45:
46: include Makefile.deps
47:
```