How to invoke the program

1. Extract the zip file.
    a. Nishimura.Yohei.P2/
        i. P2.java
        ii. b.jlex
        iii. Makefile
        iv. ErrMsg.java
        v. sym.java
        vi. *.in files (to be used in tests. see Note section.)
        vii. deps/
            1. JLex/
                a. *.class files
            2. java_cup/
                a. (files)
    b. Nishimura.Yohei.P2.pdf
2. Execute the following
    a. make
3. Expected outputs are below:

---

javac -g -cp ./deps:. sym.java
java -cp ./deps:. JLex.Main b.jlex
Processing first section -- user code.
Processing second section -- JLex declarations.
Processing third section -- lexical rules.
Creating NFA machine representation.
NFA comprised of 382 states.
Working on character
classes.::.:::::.:.:::.:....:.:.....:.....:.....::.....:..:.....:::::::.....:.....:.:.:.:::.:....:.::.....:.....:.:::::.....
NFA has 46 distinct character classes.
Creating DFA transition table.
Working on DFA states.....
Minimizing DFA transition table.
101 states after removal of redundant states.
Outputting lexical analyzer code.
javac -g -cp ./deps:. ErrMsg.java
javac -g -cp ./deps:. b.jlex.java
javac -g -cp ./deps:. P2.java

---

4. Execute the following
    a. make test
5. Expected outputs are below: (I printed out the valid tokenization for check)

```
java -cp ./deps:. P2
Start validReserved.in
1:1-4 int is tokenized.
Char.num is reset because of finding \n
2:1-5 bool is tokenized.
Char.num is reset because of finding \n
3:1-5 void is tokenized.
Char.num is reset because of finding \n
4:1-4 tru is tokenized.
Char.num is reset because of finding \n
5:1-4 fls is tokenized.
Char.num is reset because of finding \n
6:1-7 struct is tokenized.
Char.num is reset because of finding \n
7:1-8 receive is tokenized.
Char.num is reset because of finding \n
8:1-6 print is tokenized.
Char.num is reset because of finding \n
9:1-3 if is tokenized.
Char.num is reset because of finding \n
10:1-5 else is tokenized.
Char.num is reset because of finding \n
11:1-6 while is tokenized.
Char.num is reset because of finding \n
12:1-4 ret is tokenized.
Char.num is reset because of finding \n
Start validIntegerLiteral.in
1:1-4 123 is tokenized.
Char.num is reset because of finding \n
2:1-2 3 is tokenized.
Char.num is reset because of finding \n
Start validIdent.in
1:1-2 x is tokenized.
Char.num is reset because of finding \n
2:1-6 tmp_x is tokenized.
Char.num is reset because of finding \n
3:1-3 x1 is tokenized.
Char.num is reset because of finding \n
4:1-3 x_ is tokenized.
Char.num is reset because of finding \n
5:1-3 _x is tokenized.
Char.num is reset because of finding \n
```

Start validStringLiteral.in
1:1-8 "YOHEI" is tokenized.
Char.num is reset because of finding \n
2:1-12 "NICE \\, " is tokenized.
Char.num is reset because of finding \n
3:1-7 "&!99" is tokenized.
Char.num is reset because of finding \n
4:1-10 "\n char" is tokenized.
Char.num is reset because of finding \n
5:1-10 " \" \\ " is tokenized.
Char.num is reset because of finding \n
6:1-20 "\n \t \' \" \? \\" is tokenized.
Char.num is reset because of finding \n
Start validSymbol.in
1:1-2 { is tokenized.
Char.num is reset because of finding \n
2:1-2 } is tokenized.
Char.num is reset because of finding \n
3:1-2 (is tokenized.
Char.num is reset because of finding \n
4:1-2) is tokenized.
Char.num is reset because of finding \n
5:1-2 ; is tokenized.
Char.num is reset because of finding \n
6:1-2 , is tokenized.
Char.num is reset because of finding \n
7:1-2 . is tokenized.
Char.num is reset because of finding \n
8:1-3 << is tokenized.
Char.num is reset because of finding \n
9:1-3 >> is tokenized.
Char.num is reset because of finding \n
10:1-3 ++ is tokenized.
Char.num is reset because of finding \n
11:1-3 -- is tokenized.
Char.num is reset because of finding \n
12:1-2 - is tokenized.
Char.num is reset because of finding \n
13:1-2 + is tokenized.
Char.num is reset because of finding \n
14:1-2 * is tokenized.
Char.num is reset because of finding \n

15:1-2 / is tokenized.
Char.num is reset because of finding \n
16:1-2 ! is tokenized.
Char.num is reset because of finding \n
17:1-3 && is tokenized.
Char.num is reset because of finding \n
18:1-3 || is tokenized.
Char.num is reset because of finding \n
19:1-3 == is tokenized.
Char.num is reset because of finding \n
20:1-3 != is tokenized.
Char.num is reset because of finding \n
21:1-2 < is tokenized.
Char.num is reset because of finding \n
22:1-2 > is tokenized.
Char.num is reset because of finding \n
23:1-3 <= is tokenized.
Char.num is reset because of finding \n
24:1-3 >= is tokenized.
Char.num is reset because of finding \n
25:1-2 = is tokenized.
Char.num is reset because of finding \n
Start validComment.in
3:1-5 "Hi" is tokenized.
Find whitespace: CharNum+1
5:1-11 notComment is tokenized.
Char.num is reset because of finding \n
Start validWhitespace.in
1:1-4 123 is tokenized.
Find whitespace: CharNum+1
1:5-8 123 is tokenized.
Find whitespace: CharNum+1
1:9-10 3 is tokenized.
Char.num is reset because of finding \n
2:1-9 "tew tw" is tokenized.
Char.num is reset because of finding \n
3:1-8 awet_23 is tokenized.
Find whitespace: CharNum+1
3:9-16 ewaw_12 is tokenized.
Char.num is reset because of finding \n
Start invalidIntegerLiteral.in
1:1 ***WARNING*** integer literal too large; using max value

```
1:2-26 2147483647 is tokenized.
Char.num is reset because of finding \n
2:1 ***ERROR*** inproper integer literal: --25
Char.num is reset because of finding \n
Start invalidStringLiteral.in
1:1 ***ERROR*** unterminated String literal is ignored: "unterminated
Char.num is reset because of finding \n
2:1 ***ERROR*** unterminated String literal is ignored: "also \"
Char.num is reset because of finding \n
3:1 ***ERROR*** inproper char with bad escape is ignored: "backslash \
3:1 ***ERROR*** unterminated String literal is ignored: "
Char.num is reset because of finding \n
4:1 ***ERROR*** inproper char with bad escape is ignored: "bad escape \a
4:1 ***ERROR*** unterminated String literal is ignored: "
Char.num is reset because of finding \n
5:1 ***ERROR*** inproper char with bad escape is ignored: "bad escape \a
Char.num is reset because of finding \n
6:1 ***ERROR*** unterminated String literal is ignored: "unterminated string (eof)
Char.num is reset because of finding \n
Start invalidIdent.in
1:1 ***ERROR*** inproper ident starting: 1_
Char.num is reset because of finding \n
2:1 ***ERROR*** inproper ident starting: 1x
Char.num is reset because of finding \n
3:1 ***ERROR*** inproper ident starting: 1x_
Char.num is reset because of finding \n
4:1 ***ERROR*** inproper ident starting: 123x_
Char.num is reset because of finding \n
Start invalidComment.in
1:1 ***ERROR*** ignoring illegal character: #
1:2 ***ERROR*** ignoring illegal character: #
Char.num is reset because of finding \n
Start invalidSymbol.in
1:1 ***ERROR*** inproper symbol; <==
Char.num is reset because of finding \n
2:1 ***ERROR*** inproper symbol; >==
Char.num is reset because of finding \n
3:1 ***ERROR*** inproper symbol; !==
Char.num is reset because of finding \n
4:1 ***ERROR*** inproper symbol; <=======
Char.num is reset because of finding \n
5:1 ***ERROR*** inproper symbol; !=======
```

Char.num is reset because of finding \n
6:1 ***ERROR*** inproper symbol; >====
Char.num is reset because of finding \n
7:1 ***ERROR*** inproper symbol; +++++
Char.num is reset because of finding \n
8:1 ***ERROR*** inproper symbol; -----
Char.num is reset because of finding \n
9:1 ***ERROR*** inproper symbol; ***
Char.num is reset because of finding \n
10:1 ***ERROR*** inproper symbol; //
Char.num is reset because of finding \n
11:1 ***ERROR*** inproper symbol; &&&&&
Char.num is reset because of finding \n
12:1 ***ERROR*** inproper symbol; |||||
Char.num is reset because of finding \n
13:1 ***ERROR*** inproper symbol; <<<<<
Char.num is reset because of finding \n
14:1 ***ERROR*** inproper symbol; >>>>>
Char.num is reset because of finding \n
Start invalidReserved.in
1:1 ***ERROR*** ignoring illegal character: ~
Char.num is reset because of finding \n
2:1 ***ERROR*** ignoring illegal character: $
Char.num is reset because of finding \n
3:1 ***ERROR*** ignoring illegal character: %
Char.num is reset because of finding \n
4:1 ***ERROR*** ignoring illegal character: ^
Char.num is reset because of finding \n
5:1 ***ERROR*** ignoring illegal character: &
Char.num is reset because of finding \n
6:1 ***ERROR*** ignoring illegal character: |
Char.num is reset because of finding \n
7:1 ***ERROR*** ignoring illegal character: ~
7:2-3 a is tokenized.
Char.num is reset because of finding \n
8:1 ***ERROR*** ignoring illegal character: $
8:2-3 a is tokenized.
Char.num is reset because of finding \n
9:1 ***ERROR*** ignoring illegal character: %
9:2-3 a is tokenized.
Char.num is reset because of finding \n
10:1 ***ERROR*** ignoring illegal character: ^

10:2-4 ad is tokenized.
Char.num is reset because of finding \n
11:1 ***ERROR*** ignoring illegal character: &
11:2-3 g is tokenized.
Char.num is reset because of finding \n
12:1 ***ERROR*** ignoring illegal character: |
12:2-3 g is tokenized.
Char.num is reset because of finding \n
13:1 ***ERROR*** ignoring illegal character: @
Char.num is reset because of finding \n
14:1 ***ERROR*** ignoring illegal character: #
Char.num is reset because of finding \n
15:1 ***ERROR*** ignoring illegal character: \
Char.num is reset because of finding \n
16:1 ***ERROR*** ignoring illegal character: [
Char.num is reset because of finding \n
17:1 ***ERROR*** ignoring illegal character:]
Char.num is reset because of finding \n
18:1 ***ERROR*** ignoring illegal character: [
18:2 ***ERROR*** ignoring illegal character:]
Char.num is reset because of finding \n
Char.num is reset because of finding \n
Start validCode.in
1:1-7 public is tokenized.
Find whitespace: CharNum+1
1:8-13 class is tokenized.
Find whitespace: CharNum+1
1:14-16 P2 is tokenized.
Find whitespace: CharNum+1
1:17-18 { is tokenized.
Char.num is reset because of finding \n
Find whitespace: CharNum+1
2:5-11 public is tokenized.
Find whitespace: CharNum+1
2:12-18 static is tokenized.
Find whitespace: CharNum+1
2:19-23 void is tokenized.
Find whitespace: CharNum+1
2:24-28 main is tokenized.
2:28-29 (is tokenized.
2:29-30) is tokenized.
Find whitespace: CharNum+1

2:31-37 throws is tokenized.
Find whitespace: CharNum+1
2:38-49 IOException is tokenized.
Find whitespace: CharNum+1
2:50-51 { is tokenized.
Char.num is reset because of finding \n
Find whitespace: CharNum+1
Find whitespace: CharNum+1
Find whitespace: CharNum+1
5:9-15 String is tokenized.
Find whitespace: CharNum+1
5:16-23 inFiles is tokenized.
Find whitespace: CharNum+1
5:24-25 = is tokenized.
Find whitespace: CharNum+1
Char.num is reset because of finding \n
Find whitespace: CharNum+1
6:11-12 { is tokenized.
Char.num is reset because of finding \n
Find whitespace: CharNum+1
7:13-31 "validReserved.in" is tokenized.
7:31-32 , is tokenized.
Find whitespace: CharNum+1
7:33-57 "validIntegerLiteral.in" is tokenized.
7:57-58 , is tokenized.
Find whitespace: CharNum+1
7:59-74 "validIdent.in" is tokenized.
7:74-75 , is tokenized.
Find whitespace: CharNum+1
Char.num is reset because of finding \n
Find whitespace: CharNum+1
8:13-36 "validStringLiteral.in" is tokenized.
8:36-37 , is tokenized.
Find whitespace: CharNum+1
8:38-54 "validSymbol.in" is tokenized.
8:54-55 , is tokenized.
Find whitespace: CharNum+1
8:56-73 "validComment.in" is tokenized.
8:73-74 , is tokenized.
Find whitespace: CharNum+1
Char.num is reset because of finding \n
Find whitespace: CharNum+1

9:13-33 "validWhitespace.in" is tokenized.
9:33-34 , is tokenized.
Find whitespace: CharNum+1
9:35-61 "invalidIntegerLiteral.in" is tokenized.
9:61-62 , is tokenized.
Find whitespace: CharNum+1
Char.num is reset because of finding \n
Find whitespace: CharNum+1
10:13-38 "invalidStringLiteral.in" is tokenized.
10:38-39 , is tokenized.
Find whitespace: CharNum+1
10:40-57 "invalidIdent.in" is tokenized.
10:57-58 , is tokenized.
10:58-77 "invalidComment.in" is tokenized.
10:77-78 , is tokenized.
Find whitespace: CharNum+1
Char.num is reset because of finding \n
Find whitespace: CharNum+1
11:13-31 "invalidSymbol.in" is tokenized.
11:31-32 , is tokenized.
Find whitespace: CharNum+1
11:33-53 "invalidReserved.in" is tokenized.
Char.num is reset because of finding \n
Find whitespace: CharNum+1
12:11-12 } is tokenized.
12:12-13 ; is tokenized.
Char.num is reset because of finding \n
Find whitespace: CharNum+1
13:9-15 String is tokenized.
Find whitespace: CharNum+1
13:16-24 outFiles is tokenized.
Find whitespace: CharNum+1
13:25-26 = is tokenized.
Find whitespace: CharNum+1
Char.num is reset because of finding \n
Find whitespace: CharNum+1
14:11-12 { is tokenized.
Char.num is reset because of finding \n
Find whitespace: CharNum+1
15:13-32 "validReserved.out" is tokenized.
15:32-33 , is tokenized.
Find whitespace: CharNum+1

15:34-59 "validIntegerLiteral.out" is tokenized.
15:59-60 , is tokenized.
Find whitespace: CharNum+1
15:61-77 "validIdent.out" is tokenized.
15:77-78 , is tokenized.
Find whitespace: CharNum+1
Char.num is reset because of finding \n
Find whitespace: CharNum+1
16:13-37 "validStringLiteral.out" is tokenized.
16:37-38 , is tokenized.
Find whitespace: CharNum+1
16:39-56 "validSymbol.out" is tokenized.
16:56-57 , is tokenized.
Find whitespace: CharNum+1
16:58-76 "validComment.out" is tokenized.
16:76-77 , is tokenized.
Find whitespace: CharNum+1
Char.num is reset because of finding \n
Find whitespace: CharNum+1
17:13-34 "validWhitespace.out" is tokenized.
17:34-35 , is tokenized.
17:35-62 "invalidIntegerLiteral.out" is tokenized.
17:62-63 , is tokenized.
Char.num is reset because of finding \n
Find whitespace: CharNum+1
18:13-39 "invalidStringLiteral.out" is tokenized.
18:39-40 , is tokenized.
18:40-58 "invalidIdent.out" is tokenized.
18:58-59 , is tokenized.
Find whitespace: CharNum+1
18:60-80 "invalidComment.out" is tokenized.
18:80-81 , is tokenized.
Find whitespace: CharNum+1
Char.num is reset because of finding \n
Find whitespace: CharNum+1
19:13-32 "invalidSymbol.out" is tokenized.
19:32-33 , is tokenized.
Find whitespace: CharNum+1
19:34-55 "invalidReserved.out" is tokenized.
Char.num is reset because of finding \n
Find whitespace: CharNum+1
20:11-12 } is tokenized.

20:12-13 ; is tokenized.
Char.num is reset because of finding \n
Char.num is reset because of finding \n
Find whitespace: CharNum+1
22:9-12 for is tokenized.
Find whitespace: CharNum+1
22:13-14 (is tokenized.
22:14-17 int is tokenized.
Find whitespace: CharNum+1
22:18-19 i is tokenized.
Find whitespace: CharNum+1
22:20-21 = is tokenized.
Find whitespace: CharNum+1
22:22-23 0 is tokenized.
22:23-24 ; is tokenized.
Find whitespace: CharNum+1
22:25-26 i is tokenized.
Find whitespace: CharNum+1
22:27-28 < is tokenized.
Find whitespace: CharNum+1
22:29-36 inFiles is tokenized.
22:36-37 . is tokenized.
22:37-43 length is tokenized.
22:43-44 ; is tokenized.
Find whitespace: CharNum+1
22:45-47 ++ is tokenized.
22:47-48 i is tokenized.
22:48-49) is tokenized.
Find whitespace: CharNum+1
Char.num is reset because of finding \n
Find whitespace: CharNum+1
23:9-10 { is tokenized.
Char.num is reset because of finding \n
Find whitespace: CharNum+1
24:11-24 testAllTokens is tokenized.
24:24-25 (is tokenized.
24:25-32 inFiles is tokenized.
24:32-33 , is tokenized.
Find whitespace: CharNum+1
24:34-42 outFiles is tokenized.
24:42-43) is tokenized.
24:43-44 ; is tokenized.

Char.num is reset because of finding \n
Find whitespace: CharNum+1
25:11-18 CharNum is tokenized.
25:18-19 . is tokenized.
25:19-22 num is tokenized.
Find whitespace: CharNum+1
25:23-24 = is tokenized.
Find whitespace: CharNum+1
25:25-26 1 is tokenized.
25:26-27 ; is tokenized.
Char.num is reset because of finding \n
Find whitespace: CharNum+1
26:9-10 } is tokenized.
Char.num is reset because of finding \n
Find whitespace: CharNum+1
27:5-6 } is tokenized.
Char.num is reset because of finding \n
Char.num is reset because of finding \n
diff validReserved.in validReserved.out
diff validIntegerLiteral.in validIntegerLiteral.out
diff validIdent.in validIdent.out
diff validStringLiteral.in validStringLiteral.out
diff validSymbol.in validSymbol.out
diff validComment_expect.in validComment.out
diff validWhitespace_expect.in validWhitespace.out
diff invalidIntegerLiteral_expect.in invalidIntegerLiteral.out
diff invalidStringLiteral_expect.in invalidStringLiteral.out
diff invalidIdent_expect.in invalidIdent.out
diff invalidComment_expect.in invalidComment.out
diff invalidSymbol_expect.in invalidSymbol.out
diff invalidReserved_expect.in invalidReserved.out
diff validCode_expect.in validCode.out

Note

1. *.in files are below. The file name represents the test.
    a. validReserved.in, invalidReserved.in, invalidReserved_expect.in,
    b. validIdent.in, invalidIdent.in, invalidIdent_expect.in,
    c. validIntegerLiteral.in, invalidIntegerLiteral.in,invalidIntegerLiteral_expect.in
    d. validStringLiteral.in, invalidStringLiteral.in, invalidStringLiteral_expect.in
    e. validSymbol.in, invalidSymbol.in, invalidSymbol_expect.in
    f. validComment.in, validComment_expect.in, invalidComment.in, invalidComment_expect.in

g. validWhitespace.in, validWhitespace_expect.in
h. validCode.in, validCode_expect.in