



---

# Pascal-like Flex&Bison

Assignment (Lab) #4

# Index

1. Implement `pascal-like.l` and test this with in line and a test file.
2. Convert `pascal-like.bnf` to `pascal-like.y`

# 1. Implement pascal-like.l and test this with in line and a test file.

## - Result

```
iqeq126@DESKTOP-OPBS2NM:~/pascal-like1$ ls
pascal-like.bnf  pascal-like.l
iqeq126@DESKTOP-OPBS2NM:~/pascal-like1$ vi
iqeq126@DESKTOP-OPBS2NM:~/pascal-like1$ vi pascal-like.l
iqeq126@DESKTOP-OPBS2NM:~/pascal-like1$ sudo flex pascal-like.l
iqeq126@DESKTOP-OPBS2NM:~/pascal-like1$ sudo gcc -o pascal-like lex.yy.c -lf -DPRINT
iqeq126@DESKTOP-OPBS2NM:~/pascal-like1$ ls
lex.yy.c  pascal-like  pascal-like.bnf  pascal-like.l
iqeq126@DESKTOP-OPBS2NM:~/pascal-like1$ ./pascal-like
=:
Mystery cahraction
Mystery cahraction
+
Token: PLUS
:=
Token: ASSIGN
-
Token: MINUS
begin
Token: begin
end
Token: end
if
Token: if
^Z
[1]+  Stopped                  ./pascal-like
```

## - Source code

```
iqeq126@DESKTOP-OPBS2NM: ~/pascal-like1
%{
#ifdef PRINT
#define TOKEN(t) printf("Token: " #t "\n");
#else
#define TOKEN(t) return(t);
#endif
%}

%%
" ; "      TOKEN(SEMICOLON)
" := "     TOKEN(ASSIGN)
" < "     TOKEN(LESS_THAN)
" > "     TOKEN(GREATER_THAN)
" + "     TOKEN(PLUS)
" - "     TOKEN(MINUS)
begin     TOKEN(begin)
end       TOKEN(end)
if        TOKEN(if)
then      TOKEN(then)
else      TOKEN(else)
while     TOKEN(while)
do        TOKEN(do)
.         printf("Mystery cahraction\n");
%%
```

## - Explanation

1) pascal-like.l

2) flex pascal-like.l

// -> lex.yy.c

3) gcc -o pascal-like lex.yy.c -DPRINT // -> pascal-like

⇒ DPRINT 대신 두 번째 라인에 #define PRINT를 추가하여 출력할 수도 있다.

⇒ TOKEN(t) 매크로 함수에 PRINT가 정의되었을 때 Token : #t \n을 출력하고 아니면 t를 반환한다. 정의되지 않은 문자가 들어오면 Mystery character \n을 출력한다.

- testfile.file 을 통한 추가 동작 확인

testfile

```
var: Integer;  
B: Boolean;  
begin  
    if A > 10 then  
        DoSomething;  
    else  
        DoSomethingElse;  
    B := A > 10;  
    while B do  
    begin  
        DoSomething;  
    end  
end;
```

## 동작 확인 1

```
jlee126@DESKTOP-OPBS2NM:~/pascal-like1$ ./pascal-like < testfile.pas
Mystery cahraction
Mystery cahraction
Mystery cahraction
Mystery cahraction
Mystery cahraction
Mystery cahraction
Mystery cahraction
Mystery cahraction
Mystery cahraction
Mystery cahraction
Mystery cahraction
Mystery cahraction
Mystery cahraction
Mystery cahraction
Mystery cahraction
Mystery cahraction
Mystery cahraction
Mystery cahraction
Mystery cahraction
Mystery cahraction
Mystery cahraction
Mystery cahraction
Mystery cahraction
Mystery cahraction
Mystery cahraction
Mystery cahraction
Mystery cahraction
Mystery cahraction
Mystery cahraction
Mystery cahraction
Token: begin
Mystery cahraction
Token: if
Mystery cahraction
Mystery cahraction
Mystery cahraction
Token: GREATER_THAN
Mystery cahraction
Mystery cahraction
Mystery cahraction
Mystery cahraction
Token: then
```

동작확인2

iqeq126@DESKTOP-OPBS2NM: ~/pascal-like1

[illegible]

### 동작 확인3

```
iqeq126@DESKTOP-OPBS2NM: ~/pascal-like1
Token: SEMICOLON
Mystery cahraction
Mystery cahraction
Mystery cahraction
Token: ASSIGN
Mystery cahraction
Mystery cahraction
Mystery cahraction
Token: GREATER_THAN
Mystery cahraction
Mystery cahraction
Mystery cahraction
Token: SEMICOLON

Mystery cahraction
Token: while
Mystery cahraction
Mystery cahraction
Mystery cahraction
Token: do

Mystery cahraction
Token: begin

Mystery cahraction
Mystery cahraction
Mystery cahraction
Mystery cahraction
Mystery cahraction
Mystery cahraction
Mystery cahraction
Mystery cahraction
Mystery cahraction
Mystery cahraction
Mystery cahraction
Mystery cahraction
Token: SEMICOLON

Mystery cahraction
Token: end

Token: end
Token: SEMICOLON
```

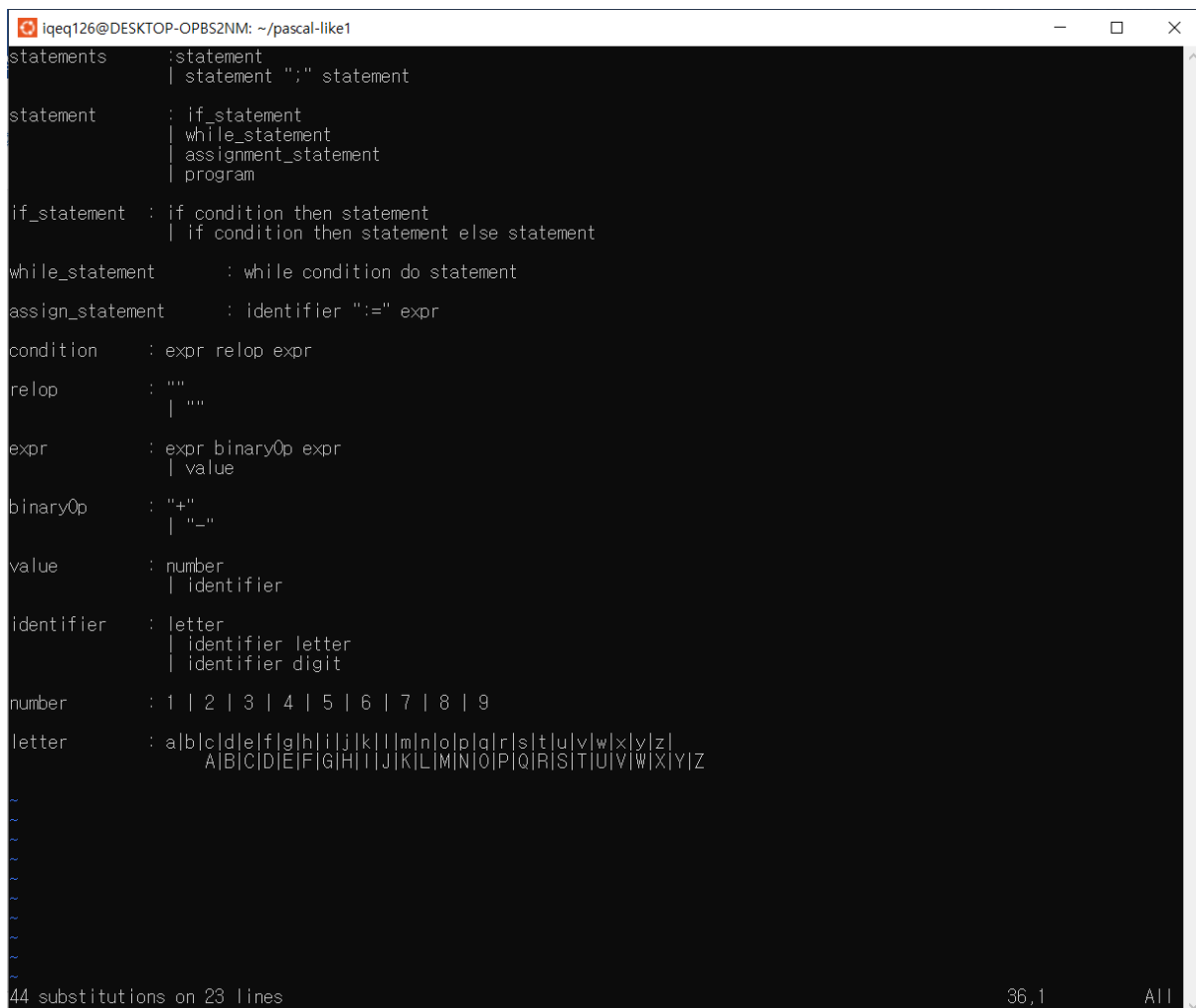
## 2. Convert pascal-like.bnf to pascal-like.y

- Copy pascal-like.bnf file to pascal-like.y

```
iqeq126@DESKTOP-OPBS2NM:~/pascal-like1$ sudo cp pascal-like.bnf pascal-like.y
iqeq126@DESKTOP-OPBS2NM:~/pascal-like1$ ls
lex.yy.c  pascal-like  pascal-like.bnf  pascal-like.l  pascal-like.y
```

- Remove every opening & closing brackets (""") in pascal-like.y
- Change bnf assign style to bison style, for example ::= -> :

'::=' -> ':'



```
iqeq126@DESKTOP-OPBS2NM: ~/pascal-like1
statements      : statement
                 | statement ";" statement

statement       : if_statement
                 | while_statement
                 | assignment_statement
                 | program

if_statement    : if condition then statement
                 | if condition then statement else statement

while_statement : while condition do statement

assignment_statement : identifier ":=" expr

condition       : expr relop expr

relop           : ""
                 | ""

expr            : expr binaryOp expr
                 | value

binaryOp        : "+"
                 | "-"

value           : number
                 | identifier

identifier       : letter
                 | identifier letter
                 | identifier digit

number          : 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

letter          : a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z|
                 A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z

~
~
~
~
~
~
44 substitutions on 23 lines 36,1 All
```

' &lt; ' - &gt; ' '

```

lqeq126@DESKTOP-OPBS2NM: ~/pascal-like1
<statements>      : <statement>
                  | <statement> ";" <statement>

<statement>       : <if_statement>
                  | <while_statement>
                  | <assignment_statement>
                  | <program>

<if_statement>     : if <condition> then <statement>
                  | if <condition> then <statement> else <statement>

<while_statement>  : while <condition> do <statement>

<assign_statement> : <identifier> ":=" <expr>

<condition>       : <expr> <relop> <expr>

<relop>           : "<"
                  | ">"

<expr>            : <expr> <binaryOp> <expr>
                  | <value>

<binaryOp>        : "+"
                  | "-"

<value>           : <number>
                  | <identifier>

<identifier>      : <letter>
                  | <identifier> <letter>
                  | <identifier> <digit>

<number>          : 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

<letter>          : a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z|
                  | A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z

%$</g>

```



```
iqeq126@DESKTOP-OPB52NM: ~/pascal-like1
statements> :statement>
| statement> ";" statement>

statement> : if_statement>
| while_statement>
| assignment_statement>
| program>

if_statement> : if condition> then statement>
| if condition> then statement> else statement>

while_statement> : while condition> do statement>

assign_statement> : identifier> ":=" expr>

condition> : expr> relop> expr>

relop> : ""
| ">"

expr> : expr> binaryOp> expr>
| value>

binaryOp> : "+"
| "-"

value> : number>
| identifier>

identifier> : letter>
| identifier> letter>
| identifier> digit>

number> : 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

letter> : a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z|
A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z

~
~
~
~
~
~
~
~
44 substitutions on 22 lines 36,1 All
```

'>' ->

[illegible]

```
iqeq126@DESKTOP-OPB52NM: ~/pascal-like1
statements      : statement
                  | statement ";" statement

statement       : if_statement
                  | while_statement
                  | assignment_statement
                  | program

if_statement    : if condition then statement
                  | if condition then statement else statement

while_statement : while condition do statement

assign_statement : identifier ":=" expr

condition       : expr relop expr

relop           : "<"
                  | ">"

expr            : expr binaryOp expr
                  | value

binaryOp        : "+"
                  | "-"

value           : number
                  | identifier

identifier      : letter
                  | identifier letter
                  | identifier digit

number          : 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

letter          : a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z|
                  A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z

~
~
~
~
~
~
~
~
~
~
44 substitutions on 23 lines                                     36,1      All
```

## relop

```
relop      : "<"  
           | ">"
```

- Change token names to matching ones with the scanner in the task #1 above**

```

lqeq126@DESKTOP-OPBS2NM: ~/pascal-like1
statements      : statement
                 | statement SEMICOLON statement

statement       : if_statement
                 | while_statement
                 | assign_statement
                 | program

if_statement    : if condition then statement
                 | if condition then statement else statement

while_statement : while condition do statement

assign_statement : identifier ASSIGN expr

condition       : expr relop expr

relop           : LESS_THAN
                 | GREATER_THAN

expr            : expr binaryOp expr
                 | value

binaryOp        : PLUS
                 | MINUS

value           : number
                 | identifier

identifier      : letter
                 | identifier letter
                 | identifier digit

number          : 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

letter          : a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z|
                 A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z

-- INSERT --
38,1
All

```