

Bonus:

1. If....else if....else

If 出現時，。

...Else if...可以出現多次，如果有任一行 if 的條件成立，parser 會判斷 if 的條件是否成立，若成立會執行其敘述，但目前只能執行一行敘述，執行完後，parser 還是會繼續執行後面的 if else 敘述，也會判斷敘述是否有 error，但 assign 值或 insert variable 等改變 symbol table 的動作就不能執行，只會將其動作印出來。

Ex.

檔案: bonus_if.go

```
var x int = 5
if(x==5)
    var c int = 10
else if(1==6)
    y = 2
else if(c > 5)
    print("c > 5")
else if(c < 5)
    print("c < 5")
else
    var a int = 10
```

執行結果

```
Create a symbol table
Insert a symbol: x
ASSIGN
IF
EQUAL ==
Insert a symbol: c
ASSIGN
ELSE IF
<Error> NOT == on line 4
<Error> Undeclare the variable "y" on line 6
ELSE IF
LARGER >
Print:  " c > 5 "
ELSE IF
<Error> NOT < on line 8
Print:  " c < 5 "
Insert a symbol: a
ASSIGN
ELSE

Total lines: 11

--- Dump the symbol table ---

NO.   Type   ID   Data
1     int    x     5
2     int    c    10
```

2. Scope

利用 array 存有幾個大的 block 和小的 block，每遇到一個左大括號，parser 就會 Print 這是第幾個 block，並且和上一個 block 比較，看新的 block 是在上一個 block 裡面，還是 not accessible。

Ex

檔案: bonus_scope.go

```
var x int = 10
{
    var y int = 5
    x++
    {
        println(x)
    }
    {
        x++
        var a float32
    }
}
{
    print(y)
    y++
    print(y)
}
}
```

執行結果

```
Create a symbol table
Insert a symbol: x
ASSIGN
Block 1
Insert a symbol: y
ASSIGN
INCREMENT
Block 2
Block 2 is in Block 1
println: 11
Block 3
Block 3 is not accessible in Block 2
INCREMENT
Insert a symbol: a
Block 4
Block 4 is not accessible in Block 1
print: 5
INCREMENT
print: 6

Total lines: 18

--- Dump the symbol table ---

NO.   Type   ID   Data
1     int    x    12
2     int    y     6
3     float   a   undefined
```