設計

因為題目要的是 interpreter，所以是每一行編譯執行一次。恩，就是每一行指令先建 AST 樹，建完之後 Evaluate，非常正規非常簡單的架構。
放屁。
教授跟我講 Compiler 的前端後端為甚麼要分開寫，然後跟我解釋了甚麼是比較正規的架構，我聽完深受啟發，所以這次作業也還是想「盡量」用這種方式。雖然很痛苦，但做完很有成就……並沒有。然後雖然架構很接近正規，但變數名稱都像是壓縮過似的……這很不好，但是沒辦法我實在太累了，就懶了。

最值得提的是 function call 的部分：

function 本身其實就可以被看作一個 AST 樹，但它又被看做一個資料型態，它要能任意移動、被定義、被存取。重點是 function 的「小 AST 樹」，要能被直接搬到程式的「大 AST 樹」call 這個 function 的地方，然後直接把小樹「嫁接」上去。當然是在 Evaluation 的時候做。
所以我的作法是，在分析 function expression 的語法時，整個 function expression 是一個 AST 樹節點，這個節點代表的就是「一個 function」，就像 num 跟 bool 的節點一樣。num 的數值存的是那個數字，bool 的數值存是或否，而 function 的數值，存的是 function body 整個 AST 樹(當然是存樹根的指標不然早爆了)。那麼 function 定義的 parameters 名稱呢？當然就是放在 function 節點的子結點了，作為這個 function 的 operand。

接下來就有兩件事，第一是拿這個 function 去命名定義，第二是 function call。
命名定義很簡單，因為都已經直接把 function 視為資料型態了，對 define statement 來說，它根本不管你 define 的是 num、bool，還是 function，反正就是直接存到變數表，通過。
function call 就很麻煩。
在我的 AST 樹上會產生一個節點，這個結點是表示一個 operation，就像+-*/><=這類的東西，只是它的 operation type 是 function call。而它的 operand，就是所有 function call 後面傳入的輸入變數，再加上 call 的 function 本身(直接接上去)。
而 evaluation 的時候，就像遇到其他 operand 一樣操作。就像+是 evaluate 所有 operand 然後相加往上傳、=就是 evaluate 所有 operand 然後相等就往上傳是不然就傳否。那 function call 就是把所有 function 以外的 operand 綁定到 function 的 parameters 然後 push 進變數的 stack 產生 function call 的 scope 然後在這個 scope 當中再加上所有 function body 定義的區域變數包含 nested function 或者 num 跟 bool 最後 evaluate function body 的計算部分然後把結果往上傳。

……沒關係你們只要知道我的程式可以過就好 (自暴自棄

編譯執行

首先進入助教們給的虛擬機，然後把 l 檔跟 y 檔還有 header 檔全部放在一起，之後對著他們念這三段咒語：

bison -d -o Final.tab.c Final.y
flex -o Final.yy.c Final.l
g++ -o Final Final.tab.c Final.yy.c -ll

然後就，哇！空間不夠用呢！
別急，退出虛擬機，然後在 Virtual Box 管理員的介面仔細看這個虛擬機的詳細資料。喔，原來是分配的記憶體太少，趕快往上調，調到 10000 MB 看看。

哇！虛擬機壞掉了！輸入密碼登不進去了！
趕快回新 eeclass 再下載一台電腦裝上去，這次試試看用 2 的次方(不要問我為甚麼我不知道)，調到 8192 MB。
這樣就可以執行了，可喜可賀 ~

我不知道到底是我運氣好，還是有甚麼特殊原因用 2 的次方調就可以。所以如果你們搞不出來也不要怪我，虛擬機我完全不懂，我只知道你們給的虛擬機本來的空間我的程式就是不夠用。

然後也不要跟我說甚麼這是我的程式沒寫好所以才要這麼多空間甚麼的……AST 樹要建起來就是這樣，題目又有這麼多不同功能，不可能根據每個功能用各種不同的 node 定義，這樣根本建不起來……總之我那天跟教授請教後，我自己得到的結論是，編譯器現在也是跟隨著其他應用的趨勢，就是犧牲一點效能，以可以輕鬆的維護、以更 robust 的架構為主要考量。
反正真的不行至少我 demo 的時候會跑給你們看一次。

成果

除了 BONUS 最後第四個，也就是 First-class Function 沒有做出來，其他全部的公開測資都過了。隱藏測資畢竟沒試過所以我不知道，但我有自己打各種語法測試，基本上都沒有問題。好歹我是盡量用最全面的架構寫的，除非隱藏測資要刻意搞人，不然應該是不會出狀況。而按你們給的文件的說法，隱藏測資應該是沒有要刻意搞人的意思……頂多是不小心搞人而已……