Compiler Project 4 Report　　R11922172 王彥傑

Part 1:

主要新增一支 codeGen.c 程式，用來產生 machine code 。
我採的 stack based 的方式處理 expression 的運算，也就是會先把 operand push 進 stack 中，之後再 pop 出來，進行運算。這樣就不用處理 register allocation 的問題，因為只要用少少的 register 就夠了，但缺點是產生的 code 會很大，此外也會有很多 memory 的讀寫。

另外也有修改 symbolTable.h ，加入 offset 和 global ，分別是記錄在 activation record 的位置，和是否為 global 變數。

semanticAnalysis.c 裡面，在 function declaration 的程式裡面， function ID Node ，原本沒有把 SymbolTableEntry 加入 AST Node 裡，我有補上。

另外也修改 semanticAnalysis.c 針對大小比較、邏輯運算結果，即使 operand 是 floating point ，也強制轉為 integer 。因為這些比較結果都會是 0 和 1 ，不是用 floating point 存放的。

另外除了題目要求以外，我還有實作下列項目：

- Variable initializations
- Multiple dimensional arrays
- Implicit type conversions (只有 int to float)

因為測試環境要求， test data 中的 main function 都改為 MAIN 。

Part 2:

我在程式中有實作 constant folding ，產生 expression 程式碼時，如果發現這個 expression 裡面在 semantic analysis 已經有把 constant 裡面的值算好了，就不會再產生程式碼去做計算。原本 semanticAnalysis.c 裡面就已經有實作好計算 constants 所組成 expression 的值了，我把它的結果拿來用。

此外，我還有製作 Dead code elimination ，如果 if 裡面的條件是 constant 或是由 constant 組成的 expression，那麼就會看 constant 的內容產生 if body 或 else body 的程式碼。