VP COMPILER CONSTRUCTION

FINAL PROJECT -SUMMARY COMPILER CONSTRUCTION

Luis Valdivia, Stefan Steininger

Assignment 0: Your Team! -Works ✔
inserting '*print((int*) "This is **Oohoo3ic** Selfie"); println();'* in main Method.

Assignment 1: Bitwise Shift Instructions -Works ✔
Implementing the 4 Bitwise shift instructions for the mips interpreter to make the selfie.c file interpret assembler code. Implemented functions:

Assignment 2: Bitwise Shift Operators (Scanning, Parsing) -Works ✔
Introduction of the two new symbols << and >> and modifying the grammar. The Parsing Function '*gr_shiftExpression()*' was inserted and adapted.

Assignment 3: **Bitwise Shift Operators (Code Generation, Self-Compilation)** -Works ✔
Replacing the original function *twoToThePowerOf* whitin the *leftshift* and *rightshift* Function by using the << and >> Symbols.

Assignment 4: Constant Folding -Works ✔
Here we have modified several parsing function to enable delayed code generation. The Attribute '*Constfold*' will pass the 'folded' expression from one parsing Function to another if the expression is foldable only then code will be emitted. Signals are set by global Variables like *isLiteralNumber* or locals like *ill (isLeftLiteral)* or *irl (isRightLiteral)*.

Assignment 5: Arrays -Works ✔
Implementing the Brackets in the Scanner then we have modified the Symbol Table. Adapting 'int symbols [37] ' . We added an array declaration (declare array, at this time is used only for local arrays). Global arrays are declared at gr_cstar. Code emission was added.

Assignment 6: Two-Dimensional Arrays – Not fully working ~
global 2D-Arrays are working *global2DArrayTest()* - unfortunately not for our 2D-Arry Symbol Table. Local 2D-Array will parse correctly, but there appear addresses instead of integer values .
We think we are overwriting the addresses of the arrays. At this time, we can't find a solution.

Assignment 7: Struct Declarations -Works ✔
We extended again the grammar, and added recognition of the symbols in the scanner and parser. Structs are declared in *gr_struct*.

Assignment 8: Struct Access – Not Working ✘
Debugger is working, we had corrected some errors, compilation is working, self compilation doesn't works, The problem might be *SynbolTableEntry - int* string* → Endless loop. The assembler in the debugger looks ok.

Assignment 9: Boolean Operators (Individual) -Works ✔
We change the grammar by moving the compare operators to gr_comparison and implemented negation in factor, gr_expression is doing the “lazy evaluation” now, the compiler is jumping if the corresponding condition is fulled.

Assignment 10: Boolean Operators (Algebra) -Works ✔
The same code works for Boolean Operatos too.

Assignment 11: Memory Management -Works ✔
Implementing Free, here is checked if within the list there is a large enough gap to add the virtual address block else the bump pointer allocates new memory.