- Authors:
  - Cameron Dunn, dunnc11@mcmaster.ca
  - Rafeed Iqbal, iqbalr8@mcmaster.ca
- Group ID on Avenue: 23
- Gitlab URL: https://gitlab.cas.mcmaster.ca/dunnc11/l-3-assembly-group-23

F1

What are 'Global Variable' as defined by the code

The code considers global variables to be any variables explicitly defined in the top level program, outside of any function definition. For any chosen variable if it is not nested, it is a global variable.

Why NOP1 instructions?

NOPn instructions are unary 'no-operation' instructions. The instruction is similar to pass in Python. It is used in the visitor TopLevelProgram.py, a visitor used in translator.py. The NOP1 instruction acts as start and end markers for the tree traversal.

Roles of Visitors and Generators

The visitors and translator use the AST library. The visit function traverses the tree, and in the visitor we override the visit_XXX functions to create instructions.

GlobalVariable.py is a visitor which traverses the AST and adds nodes where a global variable is defined to a set, and pass all others. The generator StaticMemoryAllocation.py uses the resulting set to write pep9 code which allocates memory to every global variable used in the program. This visitor and generator pair is run first.

To generate code for the rest of the program, the visitor TopLevelProgram.py is run first. It also traverses the AST, however it also adds the respective Pep9 instruction for the node to a list of instruction. The generator EntryPoint then goes through the list and writes the instructions in the list to file in order.

Limitations of given code

The given translator can only translate simple programs using its visitors and generators. Not every visit_XXX function is defined, and the ones that are defined are not optimal/functional in all cases. The code is unable to handle conditional statements, use data structures like arrays, have defined functions and call them, etc. as visitors are not defined for these node types. Some given visitors are also limited. Variables with names longer than 8 characters cannot be defined, memory is allocated using BLOCK2 instead of WORD n, no anonymous assignments, no support for constant variables, and so on.

```
add_sub.pep
         BR      program
UNIV:            .WORD 42
value:           .BLOCK 2
result:          .BLOCK 2
variable:        .BLOCK 2
program: DECI     value,d
         LDWA     value,d
         ADDA     UNIV,d
         STWA     result,d
         LDWA     3,i
         STWA     variable,d
         LDWA     result,d
         SUBA     variable,d
         STWA     result,d
         LDWA     result,d
         SUBA     1,i
         STWA     result,d
         DECO     result,d
         LDBA     '\n',i
         STBA     charOut,d
         .END
```

```
simple.pep
         BR      program
x:       .BLOCK 2
program: LDWA    3,i
         ADDA    2,i
         STWA    x,d
         DECO    x,d
         LDBA    '\n',i
         STBA    charOut,d
         .END
```