Assignment 7 - Program Slicing

Submission details: Please submit a .py file. Submit via GradeScope. If you have questions on this process, get in touch via the Slack or via email.
Due: 10/19/20

In class, we worked with a program that generates a control flow graph (CFG) for a limited subset of Python. For this assignment, transform that program into a program slicer.

Required: handle straight-line programs
Strongly encouraged: handle the if then statements we added during class
Extra super awesome: handle loops

Please support this usage:

```
python program_slicing.py filename line_number variable_name
```

Tips and Tricks

You are *not* required to follow any of the tips and tricks on the next page. Feel free to structure your program slicer however you prefer! But if you're feeling stuck, take a look at the next page for some ideas that might help you find next steps.

Tips, tricks, and thought-provoking questions!

In addition to a top-level slice function like
`slice(nodes, line_no, var_name)`
You might want a recursive helper like:
`slice_helper(node, relevant_vars, nodes_in_slice)`

What's the right point in the program to figure out the control set? Is it when we're building the CFG? Is it when we're traversing the CFG?

You'll probably want some helper functions for figuring out what's defined at a given node and what's referenced at a given node. For figuring out what variables are referenced at a given node, you'll need to traverse an AST subtree. Maybe the helper function will look something like this:

```
def referenced_at_node_helper(ast_node):
    if (type(ast_node) == ast.Constant):
        return []
    elif (type(ast_node) == ast.BinOp):
        return referenced_at_node_helper(ast_node.left) +
               referenced_at_node_helper(ast_node.right)
    ....
```

What other node types do we need to handle?