17-355/17-655/17-819: Program Analysis
Recitation Exercises
February 1, 2019 **Andrew ID: _____**

1. Define a flow function for the multiplication of two variables assigned to a third variable, ie. of the form $x := y * z$. Your flow function should be based on the **parity analysis** lattice presented in recitation. It should be as precise as possible given the analysis information available. You may define the flow function in any notation you like (e.g. mathematics, code, pseudo-code) as long as it is unambiguous.

2. Assume you had an implementation of the **zero analysis** defined in class. Explain how you would detect errors due to a division by zero. Specifically, assume a 3-address code operation of the form $x := y / z$, and describe what condition on the analysis results just before such an operation would yield (a) a definite division by zero error and (b) a possible division by zero error (e.g. in cases where the analysis is too imprecise to tell if there is definitely an error).

3. Simulate the **parity analysis** defined in recitation (using the flow function for multiplication you defined in 1 and the other flow functions defined in recitation; use the most precise version defined in recitation) on the following program using Kildall's worklist algorithm with the strongly-connected component and reverse postorder heuristics. You should produce a table as done in class. Your table should have a column for the program point, a column for the worklist, and a column for the abstract value of each variable. Each row should track the value after the execution of the corresponding statement. The rows should show how the analysis executes, examining one statement at a time:

```
1:  x := 2
2:  y := 3
3:  z := 6
4:  if y = 0 goto 8
5:  x := x * y
6:  y := y - 1
7:  goto 4
8:  y := x * z
```