Q 3.3) Faint Analysis

1) We call the set of elements for this analysis as Used Variables (UV). A variable will be termed as a used variable if either of the following conditions hold:
    - The variable must be used in a routine that obviously interacts with the outside world (eg. printing to console), is used to determine the control flow (eg. loop variables) or if it is certain that the value will be used at the end of the code segment (eg. returning a value)
    - The variable is used to define a variable that is a Used Variable

2) The direction of the analysis will be backwards. This will allow faster convergence (explained later)

3) The Transfer function is defined as:
    In[b] = {Uses*} U Out[b]

    Where Members of the Uses set are a member of the Uses set iff the variable defined by the use is a member of the UV set.*
    Additionally, uses as mentioned in the first point of section 1

    Example) In the expression,

    z = x + y

    x and y will be in the Uses set if z is in the UV set*

    In the expression,

    incrementGlobalCounterByX(x)

    x will be in the Uses set*

    In the expression,

    if(x>53)

    x will be in the Uses set*

    In the expression,

    return z

    z will be in the Uses set*

4) The meet operator is the Union operator because we need to propagate the useful variable that comes from any path

5) The ENTRY and EXIT are both initialized to the empty set which is the Bottom element for our meet operator

6) The IN and OUT are also initialized to the empty set

7) The obviously main impact would be the switch of the in and out if done forward. It doesn't have impact on the correctness as no more could be added to the UV set, it's just that at each iteration only one or so variable is to be added to the UV set, making

this iterate more times than it needs to, so this is why we would like to implement it backwards.

8) The algorithm converges because the analysis is monotonic. The Used Variable set grows towards the bottom element of the lattice (Case when all variables are used)

9) FOR each variable definition:
    IF defined variable NOT in UV:
        (expression) = FAINT