ECE 468
Problem Set 6: Dataflow analysis

Consider the following code:

```
    1: READ(x);
    2: READ(y);
L1  3: if (x > 9) goto L4
    4:   if (y > 3) goto L2
    5:     x = 3 + x;
    6:     b = y + x;
    7:     goto L3
L2  8:     y = 3 + x;
    9:     b = y + x;
L3 10:   y = x + y;
   11:   goto L1;
L4 12: WRITE(b)
   13: halt
```

1. Draw the CFG for this piece of code.

   We'll "draw" the CFG by giving the predecessor(s) and successor(s) for each statement in the program.

Statement	**Predecessor**	**Successor**
1	—	2
2	1	3
3	2, 11	4, 12
4	3	5, 8
5	4	6
6	5	7
7	6	10
8	4	9
9	8	10
10	7, 9	11
11	10	3
12	3	13
13	12	—

2. Show the results of running a *reaching definitions* analysis on this code. For each line of code, show what definitions reach that line. Assume this is the only code in the program.

   We will represent a definition by [v, n], meaning variable v was defined at line n. For each statement, we will show the GEN and KILL sets, with [v, *] meaning that all definitions of x are killed.

Statement	Predecessor	Successor	GEN	KILL
1	—	2	[x, 1]	[x, *]
2	1	3	[y, 2]	[y, *]
3	2, 11	4, 12	—	—
4	3	5, 8	—	—
5	4	6	[x, 5]	[x, *]
6	5	7	[b, 6]	[b, *]
7	6	10	—	—
8	4	9	[y, 8]	[y, *]
9	8	10	[b, 9]	[b, *]
10	7, 9	11	[y, 10]	[y, *]
11	10	3	—	—
12	3	13	—	—
13	12	—	—	—

Reaching definitions is a *forward* analysis that uses $\cup$ to merge information. That means that the two dataflow equations we will use to compute IN and OUT sets for each statement are:

$$\begin{aligned} IN(s) &= \bigcup_{t \in \mathbf{pred}(s)} OUT(t) \\ OUT(s) &= (IN(s) - KILL(s)) \cup GEN(s) \end{aligned}$$

We will iterate these equations, updating every statements IN and OUT sets, until the values stop changing. When we're done, we get:

Statement	IN	OUT
1	—	[x, 1]
2	[x, 1]	[x, 1], [y, 1]
3	[x, 1], [x, 5], [y, 1], [y, 10], [b, 6], [b, 9]	[x, 1], [x, 5], [y, 1], [y, 10], [b, 6], [b, 9]
4	[x, 1], [x, 5], [y, 1], [y, 10], [b, 6], [b, 9]	[x, 1], [x, 5], [y, 1], [y, 10], [b, 6], [b, 9]
5	[x, 1], [x, 5], [y, 1], [y, 10], [b, 6], [b, 9]	[x, 5], [y, 1], [y, 10], [b, 6], [b, 9]
6	[x, 5], [y, 1], [y, 10], [b, 6], [b, 9]	[x, 5], [y, 1], [y, 10], [b, 6]
7	[x, 5], [y, 1], [y, 10], [b, 6]	[x, 5], [y, 1], [y, 10], [b, 6]
8	[x, 1], [x, 5], [y, 1], [y, 10], [b, 6], [b, 9]	[x, 1], [x, 5], [y, 8], [b, 6], [b, 9]
9	[x, 1], [x, 5], [y, 8], [b, 6], [b, 9]	[x, 1], [x, 5], [y, 8], [b, 9]
10	[x, 1], [x, 5], [y, 1], [y, 8], [y, 10], [b, 6], [b, 9]	[x, 1], [x, 5], [y, 10], [b, 6], [b, 9]
11	[x, 1], [x, 5], [y, 10], [b, 6], [b, 9]	[x, 1], [x, 5], [y, 10], [b, 6], [b, 9]
12	[x, 1], [x, 5], [y, 1], [y, 10], [b, 6], [b, 9]	[x, 1], [x, 5], [y, 1], [y, 10], [b, 6], [b, 9]
13	[x, 1], [x, 5], [y, 1], [y, 10], [b, 6], [b, 9]	[x, 1], [x, 5], [y, 10], [b, 6], [b, 9]

3. Show the results of running a *liveness* analysis on this code. For each line of code,

show what variables are live *out* for that line (i.e., what variables are live immediately after that line would execute)

As before, we start by constructing the GEN and KILL sets for each statement:

Statement	**Predecessor**	**Successor**	**GEN**	**KILL**
1	—	2	—	x
2	1	3	—	y
3	2, 11	4, 12	x	—
4	3	5, 8	y	—
5	4	6	x	x
6	5	7	y, x	b
7	6	10	—	—
8	4	9	x	y
9	8	10	x, y	b
10	7, 9	11	x, y	y
11	10	3	—	—
12	3	13	b	—
13	12	—	—	

Liveness is a *backwards* analysis that uses $\cup$ to merge together information, so the equations for IN and OUT are:

$$\begin{aligned} IN(s) &= (OUT(s) - KILL(s)) \cup GEN(s) \\ OUT(s) &= \bigcup_{t \in \mathbf{succ}(s)} IN(t) \end{aligned}$$

When we iterate these equations, we get:

Statement	**IN**	**OUT**
1	b	b, x
2	b, x	b, x, y
3	b, x, y	b, x, y
4	x, y	x, y
5	x, y	x, y
6	x, y	b, x, y
7	b, x, y	b, x, y
8	x	x, y
9	x, y	b, x, y
10	b, x, y	b, x, y
11	b, x, y	b, x, y
12	b	—
13	—	—