Result of Liveness calculation:

1	(program(v w x y z t.1 t.2)	
		{}
2	(movq (int 1) (var v))	
		{v}
3	(movq (int 46) (var w))	
		{v, w}
4	(movq (var v) (var x))	
		{w, x}
5	(addq (int 7) (var x))	
		{w, x}
6	(movq (var x) (var y))	
		{w, x, y}
7	(addq (int 4) (var y))	
		{w, x, y}
8	(movq (var x) (var z))	
		{w, y, z}
9	(addq (var w) (var z))	
		{y, z}
10	(movq (var y) (var t.1))	
		{t.1, z}
11	(negq (var t.1))	
		{t.1, z}
12	(movq (var z) (var t.2))	
		{t.1, t.2}
13	(addq (var t.1) (var t.2))	
		{t.2}
14	(movq (var t.2) (reg rax)))	
		{}

Algorithm:

If instruction I_k is:

- A move: `(movq s d)`, then add the edge (d, v) for every $v \in L_{\text{after}}(k)$ unless $v = d$ or $v = s$.
- **Not** a move but some other instruction such as `(add s d)`, then add the edge (d, v) for every $v \in L_{\text{after}}(k)$ unless $v = d$.
- Of the form `(callq` *label*`)`, then add an edge (r, v) for every caller-save register r and every variable $v \in L_{\text{after}}(k)$.

Graph: