Problems 4: Liveness analysis

1. In the following code (a sequence of quadruples), show which variables are live after each quadruple. You can assume that no variables are live at the end.

```
    w = 0
    x = 1
    y = 2
L1: z = y + 3
    w = w + z
    y = x
    y = z + 4
    if (y < 10) goto L1
    write(w)
```

1st iteration:

```
                          Live-out
    w = 0                 {w}
    x = 1                 {x,w}
    y = 2                 {x,y,w}
L1: z = y + 3             {x,z,w}
    w = w + z             {x,z,w}
    y = x                 {z,w}
    y = z + 4             {y,w}
    if (y < 10) goto L1   {w}
    write(w)
```

2nd iteration:

```
                          Live-out
    w = 0                 {w}
    x = 1                 {x,w}
    y = 2                 {x,y,w}
L1: z = y + 3             {x,z,w}
    w = w + z             {x,z,w}
    y = x                 {x,z,w}
    y = z + 4             {x,y,w}
    if (y < 10) goto L1   {x,y,w}
    write(w)
```

2. Show the interference graph for the program of Q1, explaining how you construct it.

Edges are added whenever a variable v is defined, between v and every live-out variable, except a in the case of a move instruction v=a.

Graph contains edges x-y, x-z, w-x, y-z, w-y, w-z (fully connected).

3. Without coalescing nodes, use any graph colouring algorithm to colour the interference graph. Show how this can be used to allocate the program's variables to registers.

A different colour is needed for each variable:

```
Colour w 1
Colour x 2
Colour y 3
Colour z 4
```

Need 4 registers. E.g., use R0 for w, R1 for x, R2 for y, R3 for z.