Optimising Compilers 2012–2013
Exercise Sheet 1*

The purpose of this exercise sheet is to practise *data flow optimisations*.

1. (a) Perform a live variable analysis on the following function showing the *live-in* set for each statement.

```
int f(int x, int y) {
        int z = y+y;
        y = x+1;
        x = x-y;
        z = z+y;
        x = x+1;
        y = y+x;
        z = x+1;
        y = y+z;
        return y;
}
```

   (b) Based on your LVA re-write the function `f` removing any *dead code* from the function.

   (c) Simplify the function as much as you can.

2. (a) Consider the following function written in C:

```
int g(int x) {
        int z = p(x);
        int y = q(x);
        return y;
}
```

   Is there any dead code in this function, and if so where?

   (b) Given that the operation of `p` and `q` are unknown, or their analysis is undecidable, comment on the safety of performing a *dead code elimination* phase.

Please also complete the following past exam questions:

- 2005 Paper 8 Question 7 (just part (a))
- 2006 Paper 8 Question 8 (part (c) is optional)
- 2011 Paper 7 Question 13

Past exam questions can be found at:
`http://www.cl.cam.ac.uk/teaching/exams/pastpapers/t-OptimisingCompilers.html`.

*Many thanks to Dominic Orchard, Tomas Petriček and Raoul Urma for examples in Exercise Sheets 1–4.