Exercises on Analysis and Optimisation

1. Consider the following program.

```
int foo(int e) {
  int a, b, c, d;
  a = 1;
  b = 2;
  do {
    c = a + b;
    d = c - a;
    while (e != 0) {
      d = b & d;
      if (d == 0) break;
      d = a + b;
      e = e + 1;
    }
    b = a + b;
    e = c - a;
  } until (b != 0);
  a = b & d;
  b = a - d;
  return b;
}
```

   (a) Convert the program into 3-address code. You may use any reasonable 3-address instructions.

   (b) Represent your 3-address program as a flow graph of basic blocks.

   (c) Compute the available expressions at the entry and exit of each basic block. Does your analysis suggest any opportunity for common subexpression elimination? If so, perform this optimisation.

   (d) Using your optimised flow graph, compute the reaching definitions at the entry and exit of each basic block. Does your analysis suggest any opportunity for constant propagation? If so, perform this optimisation.

[0] I'd like to thank Jacky Jiang for finding and fixing some errors in an earlier version of this document.

(e) Using your optimised flow graph, compute the live variables at the entry and exit of each basic block. Does your analysis suggest any opportunity for dead code elimination? If so, perform this optimisation.