322 Compilers Assignment: liveness-test
Register allocation, liveness testing

Due Thursday April 12th, noon

Your job: Design test cases for the liveness phase of your compiler as a pair of files, an L2 input function and the expected output.

This is the shape of the liveness function:

```
liveness : (i ...) -> ((in (var ...) ...)
                       (out (var ...) ...))
```

The `liveness` function accepts an L2 function (as a list of instructions), and returns the “in” and “out” sets for each instruction, as a sequence of variables and registers. The variables and registers in each list must be sorted alphabetically, and each sequence of variables and registers must correspond to an instruction in the input function, i.e., in the same order.

The ’var’s should include all of the variables and registers mentioned in the program, except esp and ebp.

The liveness function should be wrapped up into a script that accepts a filename naming a file that contains the arguments in the file. The script should write their answers to stdout. For example, if the file `f.L2f` contains:

```
((x <- 1) (eax += x))
```

Then this this transcript shows how your script should behave:

```
% liveness f.L2f
((in (eax) (eax x)) (out (eax x) ()))
```

Your scripts must run on the t-lab machines (under linux).

Hand in your assignment by sending email with the subject `liveness-test` to `robby@eecs.northwestern.edu`. The email should include an attachment named *`name`*`.liveness-test.tar.gz`. The *`name`* should be your last name in all lowercase letters unless you are pair programming, in which case it should be both last names in alphabetical order, separated by `+`. For example, if Robert Jordan and Brandon Sanderson were pair programming and handing in this assignment, they’d send in a tarfile named `jordan+sanderson.liveness-test.tar.gz`

The attachment must contain a single directory named `liveness-test` containing the test cases.

The input files should use the suffix `.L2f` and the correct answers should use the suffix `.lres`.