322 Compilers Assignment: `graph-test`

Your job: Design (at least) 15 test cases for graph coloring and graph construction as a pair of files, an L2 input function and the expected output. This is the shape of the coloring function:

```
graph-color : (i ...) -> ((var var ...) ...)
                         ((var reg) ...) or #f
```

The `graph-color` function accepts an L2 function (as a list of instructions), and returns two values. The first must be a graph to color and the second must either be a coloring for that graph or `#f`, indicating the algorithm failed to color the graph.

The graph should be printed out as an adjacency table, mapping each variable to its neighbors. Sort each sequence of neighbors alphabetically by the name of the variable and sort the entire graph by the names of the variables for each node. The graph should include all of the registers except `rsp`.

The coloring should be printed out as a sequence of pairs, mapping each non-register variable to a register (color), sorted by the name of the variable. If all of the verticies in the graph printed as the first result have fewer than 16 neighbors, then the result must not be `#f`. Otherwise, it should only be `#f` when the graph cannot be colored (but the checker doesn't check this aspect).

The graph-color 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:

```
(:f 0 0 (x <- 1) (rax += x) (return))
```

Then this transcript shows how your script might behave (the `r10` in the last line might be a different register):

```
% graph-color f.L2f
((r10 r11 r12 r13 r14 r15 r8 r9 rax rbp rbx rcx rdi rdx rsi)
 (r11 r10 r12 r13 r14 r15 r8 r9 rax rbp rbx rcx rdi rdx rsi)
 (r12 r10 r11 r13 r14 r15 r8 r9 rax rbp rbx rcx rdi rdx rsi x)
 (r13 r10 r11 r12 r14 r15 r8 r9 rax rbp rbx rcx rdi rdx rsi x)
 (r14 r10 r11 r12 r13 r15 r8 r9 rax rbp rbx rcx rdi rdx rsi x)
 (r15 r10 r11 r12 r13 r14 r8 r9 rax rbp rbx rcx rdi rdx rsi x)
 (r8 r10 r11 r12 r13 r14 r15 r9 rax rbp rbx rcx rdi rdx rsi)
 (r9 r10 r11 r12 r13 r14 r15 r8 rax rbp rbx rcx rdi rdx rsi)
 (rax r10 r11 r12 r13 r14 r15 r8 r9 rbp rbx rcx rdi rdx rsi x)
 (rbp r10 r11 r12 r13 r14 r15 r8 r9 rax rbx rcx rdi rdx rsi x)
 (rbx r10 r11 r12 r13 r14 r15 r8 r9 rax rbp rcx rdi rdx rsi x)
 (rcx r10 r11 r12 r13 r14 r15 r8 r9 rax rbp rbx rdi rdx rsi)
 (rdi r10 r11 r12 r13 r14 r15 r8 r9 rax rbp rbx rcx rdx rsi)
 (rdx r10 r11 r12 r13 r14 r15 r8 r9 rax rbp rbx rcx rdi rsi)
 (rsi r10 r11 r12 r13 r14 r15 r8 r9 rax rbp rbx rcx rdi rdx)
 (x r12 r13 r14 r15 rax rbp rbx))
((x r10))
```

The server uses `check-coloring` (from `322-interps.tar.gz`) to make sure your colorings are sensible. Use it to check your colorings locally before submitting.

Hand in your assignment by uploading it to the server at `http://penghu.eecs.northwestern.edu:8123/`. The uploaded file should be a gzipped tar file named *name*`.graph-test.tar.gz`. The *name* should be your family name in all lowercase letters (except for the names He, Liu, or Wang, see below) unless you are pair programming, in which case it should be both family names in alphabetical order, separated by `+`. If your name has any non-alphabetic characters, remove them first. For example, if Conan O'Brien and Shawn Knowles-Carter were pair programming and handing in this assignment, they'd send in a tarfile named `obrien+kowlescarter.graph-test.tar.gz`. If your family name is He, Liu, or Wang, then include your first name as well, but also without any spaces. For example, if your name is Liu Bolin, then use `bolinliu` as your name. And if Bolin and and Shawn team up, they'd submit `bolinliu+kowlescarter.graph-test.tar.gz`.

The file must contain a single directory named `graph-test` containing the test cases.

The input files should use the suffix `.L2f` and the correct answers should use the suffix `.gres`. Your scripts must run on the t-lab machines (under linux).