

## Assignment #9a

### Compiling and Debugging Fortran Programs

Congratulations! You've inherited another program written by someone else. This one is buggy and broken.

- 1) Download the program `letters.f` into your workspace, and then unpack it as follows:

```
tar xvzf ex9a.tgz  
cd ex9a
```

- 2) Create a “make” file to help compile your program, just as you did in Assignment #7.
- 3) Use “make” to compile your program. Wow, there are a lot of errors! The Fortran compiler tells you the line number for each error. Open the program code in your favorite editor, and look for the first syntax error. Fix the error, save the code, and try compiling again. Continue this process until the file compiles cleanly.
- 4) Now, try to run the program. If it's working correctly, you should see something like this:

```
make  
gfortran -g letters.c -o letters  
./letters  
Type in a sentence:  
Hello, world!  
Statistics:  
2 words  
  
Letter d: 1  
Letter e: 1  
Letter h: 1  
Letter l: 3  
Letter o: 2  
Letter r: 1  
Letter w: 1
```

But this program is buggy, so when you run it, it reports the wrong number of words and the wrong letters.

- 5) Use `gdb` to step through the execution of the code and figure out all of the problems. Keep fixing the program until it works correctly.

#### Lessons Learned:

- A program must be compiled before running it, after each change to the source code.
- In Fortran, the leading spaces on each line are important, and the characters in the first columns mean something.
- Even if a program compiles correctly, it doesn't necessarily run correctly.
- You can't tell what's happening inside a program unless you use `gdb` to step through it.