INTRODUCTION TO COMPILING: HELLO WORLD!

The computer does not understand directly the contents of hello.c since the language you have used, C, is a "high level language". You need to compile the program. A compiler is a program which translates your high level C program into "machine language" which the computer can more readily understand. To compile, type

```
gcc hello.c
```

At this point you may get error messages if you have typos in your program. Try to find them yourself, but please raise your hand if you need help. Once your program compiles successfully, you can type the linux ls command. You should see a file called 'a.out' which is an executable that the compiler created from your C program. You can run it by typing

```
./a.out
```

The computer should type "Hello, world" to your screen. Please raise your hand so we can come and help if this did not work out.

It is much better to tell the computer to give your executable a specific name than to let the computer call it "a.out". For one thing, the name "a.out" provides no clue that it is the executable associated with "hello.c". For another, you will want to save many executables and not have new ones write over the old one, which is what will happen if you tell the compute to call them all "a.out". So let's recompile the program giving it a better name:

```
gcc -o hello.e hello.c
```

The -o tells the compiler to name the executable "hello.e" rather that "a.out". You can name it something else, but it is useful to call it "hello.e" since then it is immediately apparent that it is the partner of "hello.c". As before, you can run your program:

```
./hello.e
```

If you type ls you should now see three files: "hello.c", "a.out", and "hello.e". Since we do not need "a.out", let's learn another useful linux command. Type

```
rm a.out
```

rm is the linux abbreviation for remove. Type ls again and verify that a.out has disappeared.