

LLVM tips

Compile C source to LLVM IR (generates `foo.ll` text file):

```
clang -S -emit-llvm foo.c
```

Compile LLVM IR to x86 ASM (generates `foo.s` text file):

```
llc foo.ll
```

Translate x86 ASM to binary object code (generates `foo.o`):

```
as foo.s -o foo.o
```

Link object code with C library (generates `a.out`, a binary executable).

```
clang foo.o
```

Yes, we're invoking the C compiler, but asking it just to do the linking. It's possible to invoke the linker directly (the command is `ld`) but that requires manually specifying a bunch of system-dependent libraries. It's much easier to ask `gcc` to do it.

Some other tricks: instead of compiling and linking down to native code, we can just run the LLVM interpreter directly:

```
lli foo.ll
```

See also: [IR for collatz program](#)