# C Review Recitation Handout

Monday, Feb 19 | Wednesday, Feb 21

## Activity 0: Reading `man` pages

- Either type `man getopt` on your Terminal or Google "man getopt".
- `int getopt(int argc, char * const argv[], const char *optstring);`

What does `getopt` do?

Describe the parameters that `getopt` takes:

- `int argc`:
- `char * const argv[]`:
- `const char *optstring`:

What does `getopt` return?

Other specifications about `getopt`:

Note: use this template to understand other C functions/man pages as well!

## Activity 1: `getopt_example.c`

In this first activity, we have given you a file called `getopt_example.c`. Your task is to figure out what this program does!

What does `getopt_example.c` do?

How do you get the program to process arguments (i.e. formatting specifics?)

What does the `-v` argument do? The `-n` argument?

## Activity 2: `pyth_solver.c`

For this next activity, you will be writing a Pythagorean triple solver in C, using getopt!

Note: this is a large activity for a few minutes of class time. Our hope is that if you don’t completely understand getopt, you can return to this activity, do it, and then understand getopt much better before cachelab.

Your program should:

- Take in arguments with a, b, c tags and determine if the numbers inputted are a Pythagorean triple.
- Error check on: number of arguments/validity of arguments (should exit on invalid arguments)
- Invalid arguments: too few/too many arguments, negative args
- Verbose mode: if verbose mode is enabled, output a^2, b^2, c^2

You will write your solution in `pyth_solver.c`.

## Compiling and running your tests on your solver

To make your solver, type:

```
$ make clean
$ make pyth_solver
```

To manually run tests on the solver, type:

```
./pyth_solver (ARGUMENTS)
```

To run the staff-given tests on your solver, type:

```
./run_tests
```

IMPORTANT: every time you edit your solver, you need to type:

```
$ make clean
```

After this, the executables are erased and then you can recompile with make.

## Additional Commands

To compile all files, type:

```
$ make clean
$ make
```

To compile just the example, type:

```
$ make getopt_example
```