# getopt warmup

This warmup serves as a tutorial in the use of `getopt()`.

All C and C++ programs (and those in many other languages) have the opportunity to receive and make use of arguments from the command line. These are communicated to the process via the arguments to its `main()` or equivalent function.

```
int main(int argc, char **argv)
```

`int argc` contains the number of non-null entries in the array `argv`.

`argv` is a null terminated array of pointers to C strings.

`argc` is never (correctly) less than 1 and `argv[0]` always points to something (which should be, by convention, the name of the program being executed).

```
$ ls
```

produces an `argc` of 1 and `argv[0]` will point to `/bin/ls` or something similar.

```
$ ls -l
```

produces an `argc` of 2. `argv[0]` will point to `/bin/ls` or something similar. `argv[1]` will point to `-l`.

`getopt()` is an easy way to parse command line arguments. The code contains in its comments a fairly thorough tutorial.

## Assignment

You are to take a screenshot of you building and executing the provided program with the addition of two command line options.

| Option | Argument | Action |
|---|---|---|
| U | string | prints the string |
| n | integer | prints the integer squared |

Your screenshot should show the above two options in use.

And, it should show a second invocation demonstrating the `-h` option to include help for the two new options.

You can make use of the provided `makefile` to build the program.

The program must be built in WSL on Windows or in the terminal on the Mac.

Submit the screenshot as a `jpg` file to Schoology.