CS253 HW0: Light Bulb Joke

Purpose

The purpose of this assignment is to make sure that you can follow instructions, login to a Linux system, compile a C++ program, and check in homework. This way, you're sure that your login and password work *before* you need them for homework #1. This assignment is **not** optional.

Description

In this assignment, you will write a program called `hw0`. It will answer the question, "How many CU students does it take to change a light bulb?"

Where's the GUI?

In this class, we build software using `cmake` and `make`. We turn in a `tar` file. You don't really need to know much about any of those programs—we supply a file `CMakeLists.txt` that is the data file for `cmake`, which produces a a file `Makefile`, the data file for `make`. All that you need to do is type `cmake .` once, and `make` every time afterwards. `make` will compile your code (if it can) and create an `hw0.tar` file for you to turn in when you're finished.

"Yeah, but where's the GUI? How do I edit my code?" Edit it any way you want. Use of a GUI or text editor is not required in this class. If you need help using a text editor, come to me for help.

Sample Build

In the following example, what you type looks like this. The percent sign, "%", is my shell prompt. My `CMakeLists.txt` is shown as an example. You may copy it verbatim, or create your own, as long as it meets the requirements below.

```
% cat CMakeLists.txt
cmake_minimum_required(VERSION 3.14)

# Using -Wall is required:
add_compile_options(-Wall)

# These compile flags are highly recommended, but not required:
add_compile_options(-Wextra -Wpedantic)

# Optional super-strict mode:
add_compile_options(-fmessage-length=80 -fno-diagnostics-show-option)
add_compile_options(-fstack-protector-all -g -O3 -std=c++14 -Walloc-zero)
add_compile_options(-Walloca -Wctor-dtor-privacy -Wduplicated-cond)
add_compile_options(-Wduplicated-branches -Werror -Wfatal-errors -Winit-self)
add_compile_options(-Wlogical-op -Wold-style-cast -Wshadow)
add_compile_options(-Wunused-const-variable=1 -Wzero-as-null-pointer-constant)

# add_compile_options must be BEFORE add_executable.

# Create the executable hw0 from the source file main.cc:
add_executable(hw0 main.cc)

# Create a tar file every time:
add_custom_target(hw0.tar ALL COMMAND tar cf hw0.tar main.cc CMakeLists.txt)
% cmake .
… cmake output appears here …
% make
… make output appears here …
% ./hw0
Three—one to hold the light bulb and two to debate
whether an LED or a CFL bulb harms the environment more!
```

Testing

It is **essential** that you test what you've turned in (the `hw0.tar` file). Here's an easy way, where `%` is my prompt:

```
% rm -rf testdir
% mkdir testdir
% cd testdir
% tar -xvf ../hw0.tar
% cmake .
% make
% ./hw0
% cd ..
```

That's how the TA will test your program (though they will use a different directory name & location). If this doesn't compile your program, then you get ***NO*** points.

Requirements

- You may not use any external programs via `system()`, `fork()`, `popen()`, `execl()`, `execvp()`, etc.
- You may not use C-style I/O facilities, such as `printf()`, `puts()`, or `putc()`.
  - Instead, use `cout`.
- You may not use dynamic memory via `new`, `delete`, `malloc()`, `calloc()`, `realloc()`, `free()`, `strdup()`, etc.
  - It's ok to *implicitly* use dynamic memory via containers such as `string` or `vector`.
- The output must end with a newline.
  - Newlines do not separate lines—newlines *terminate* lines. That means that the last line ends with a newline, as all lines do.
- We will compile your program like this: `cmake .; make`
  - If that generates warnings, you will lose a point.
    - OK, just half a point, since this is a one-point assignment.
  - If that generates errors, you will lose *all* points.
- There is no automated testing/pre-grading/re-grading.
  - Test your code yourself. It's your job.
  - Yes, even if you only change it a little bit.
  - Yes, even if all you do is add a comment.
- You will not be graded on the quality of your humor.
  - However, if you duplicate my example joke, you will lose points.
  - That's because plagiarism is bad.
  - You can't copy, even if you do give credit—I require original work.
- Turn in a `tar` file called `hw0.tar` containing `CMakeLists.txt` and a single C++ source file, `main.cc`.
  - Yes, `.cc`, not `.cp`, `.cxx`, `.cpp`, `.CPP`, `.c++`, or `.C`.
- Your `CMakeLists.txt` must use *at least* `-Wall` *every* time `g++` runs.

How to submit your homework:

Use web checkin, or Linux checkin:

```
~cs253/bin/checkin HW0 hw0.tar
```

How to receive *negative* points:

Turn in someone else's work.