CIS*2750
Assignment 1 test harness instructions

Important notes:
- Make sure you compile and run all code on linux.socs.uoguelph.ca!

1 Grading rubric

Functions (graded using an automated test harness):			90%
- `createCalendar()`:		70%	
- Correct handling of valid files and arguments	45%		
- Correct handling of invalid files and arguments	25%		
- `deleteCalendar()`:		8%	
- `printCalendar()`:		7%	
- `printError()`:		5%	

Sub-Total: 90%

Correct `makefile` (has all four rules, creates correctly named files): 10%
- `make list` creates a shared library `liblist.so` in `assign1/bin`
- `make parser` creates a shared library `libcal.so` in `assign1/bin`
- `make` or `make all` creates `liblist.so` and `libcal.so` in `assign1/bin`
- `make clean` removes all `.o` and `.so` files

Total: 100%

Additional deductions:

- Any compiler warnings:	-15%
- Any memory leaks:	-15%
- Any memory errors other than leaks, e.g. under-allocating memory, using uninitialized pointers, etc.:	-15%
- Incorrect directory structure:	-5%
- Incorrect output filenames created by makefile:	-5%
- Any additional failures to follow submission instructions:	-5%

Any compiler errors: automatic grade of zero (0) on the assignment.

2. Test harness instructions

Running the main test harness

The harness tests iCalendar creation/deletion and the two print... functions. It also tests error handling. It does not test for memory leaks - those scripts are separate (see below). As a result, iCalendar deletion tests only really test for segfaults and other crashes during deletion.
We don't want to rely on potentially broken makefiles/static libs, so the test harness only uses student .h and .c files.

The test harness directory structure is:
- `bin` - will contain all executable files
- `src` - contains test cases. Do not modify these in any way.
- `include` - contains test harness headers, as well as `LinkedListAPI.h` and `CalendarParser.h`. Do not modify these in any way.
- `studentCode` - student `.c` files go here.
- `studentInclude` - student `.h` files go here
- `testFiles` - contains various broken and valid iCalendar files

Copy your `.c` and `.h` files (except `CalendarParser.h`. and `LinkedListAPI.h`) into the appropriate directories, and run the harness.

Running the test harness

- Place your `.c` files into `studentCode`.
- Place your additional headers into `studentInclude` - i.e. everything except `CalendarParser.h`. and `LinkedListAPI.h`. **Do not** use you `CalendarParser.h` and `LinkedListAPI.h`!
- Type `make clean` before compiling to clear the old data.
- Type `make calTestA1` to compile and execute the harness. You will see the compilation feedback and, if the assignment compiles, the test harness results.

The output of the test harness contains all the information about the passed/failed tests, as well as the total score (out of 90).

Checking for memory leaks

- Compile and run by typing `make leakTestA1`

This will execute valgrind 12 times. There are 6 tests first create/delete (including 2 where create is expected to return an error code due to an invalid file), then 4 for create/print/delete. Each test must show
`" in use at exit: 0 bytes in 0 blocks".`

If there are more than 0 bytes at exit - i.e. if there are leaks - you will see something like:

```
LEAK SUMMARY:
==31556==    definitely lost: 112 bytes in 1 blocks
==31556==    indirectly lost: 20,929 bytes in 681 blocks
==31556==      possibly lost: 0 bytes in 0 blocks
==31556==    still reachable: 0 bytes in 0 blocks
==31556==         suppressed: 0 bytes in 0 blocks
```

Checking for memory errors

- Compile and run by typing `make memErrTestA1`

This will execute valgrind 12 times. There are 6 tests first create/delete (including 2 where create is expected to return an error code due to an invalid file), then 4 for create/print/delete. Each test must show

```
==14173== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
==14173== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```

Again, the unfortunately the output will be super-messy (especially with all the REDIR messages), so you will have to be careful when reading it.