

Test Harness

Instructions

- Get the data files (all .csv files) and harness folder from courselink under Assignment 3 section.
- Please use linux server only.

Have an alternate IDE installed other than no machine to avoid any trouble submitting A3 unlike A2.

NOTE:

- This harness only works if you have all the functions implemented in their own file.
- This harness is especially for some functions (read the output below).
- For delete functions, you should test it using, `playPlaylist()` or `valgrind`.
- You don't need to submit any files related to harness. For more information, see *Deliverables* on A3_expected_behaviour on courselink.
- When you run harness, make sure you must comment out everything you have in makefile before adding those provided lines below.

Structure

Harness folder and *.csv files should be present in the working directory.

cis2500A3

```
|----src
|----include
|----makefile
|----harness
|----bin
|----*.csv files (* indicates all files with .csv extension)
```

Building Makefile

Add the following lines to your makefile to run the harness. It will generate an executable in the bin folder called test. Run command `./bin/test` after *make*.

```

CC= gcc
CFLAGS = -Wall -std=c99 -w

bin/test: TestCode.o unity.o TestCodeRunner.o harness/unity.h
harness/CodeUnderTest.h
    $(CC) $(CFLAGS) -o bin/test TestCode.o unity.o TestCodeRunner.o
-lm

TestCode.o: harness/TestCode.c unity.o harness/unity.h
    $(CC) $(CFLAGS) -c harness/TestCode.c

unity.o: harness/unity.c harness/unity.h
    $(CC) $(CFLAGS) -c harness/unity.c

TestCodeRunner.o: harness/TestCodeRunner.c harness/unity.h
    $(CC) $(CFLAGS) -c harness/TestCodeRunner.c

clean test:
    rm *.o bin/test

```

The output for the code that passes all tests will look like this:

```

dkumar07@linux-05:~/cis2500A3$ ./bin/test
===== RUNNING TESTS =====

***** Begin Testing *****

Test 'test_createPlayList' PASSED
Test 'test_createPlayList1' PASSED
Test 'test_createPlayList2' PASSED
Test 'test_createPlayList3' PASSED
Test 'test_addNewSong' PASSED
Test 'test_addNewSong1' PASSED
Test 'test_addNewSong2' PASSED
Test 'test_addNewSong3' PASSED
Test 'test_playSongGivenId' PASSED
Test 'test_playSongGivenId1' PASSED
Test 'test_playSongGivenId2' PASSED

```

```
Test 'test_playSongGivenId3' PASSED
Test 'test_playSongGivenName' PASSED
Test 'test_playSongGivenName1' PASSED
Test 'test_playSongGivenName2' PASSED
Test 'test_playSongGivenName3' PASSED
Test 'test_countNotesInASong' PASSED
Test 'test_countNotesInASong1' PASSED
Test 'test_countNotesInASong2' PASSED
Test 'test_countNotesInASong3' PASSED
```

```
***** Testing Ends *****
```

```
=== TEST SUMMARY ===
```

```
Tests Run: 20
```

```
Tests Passed: 20
```

```
Tests Failed: 0
```

```
dkumar07@linux-05:~/cis2500A3$
```

These are not all the test cases which we will be testing your code on. This is provided only to make your life easier and pass most of the cases to keep you on a safer side while grading. All the Best!