The instructions say:

"The make test command runs all your tests on your implementations. You should provide sufficient testing, especially testing the boundaries of your data structures. Examples of boundary testing are: "

Does this mean we should name our executable "test"? Or is there a make test command that I'm not finding when I search for it.

---

Subject: Re: adding "Make Test" command to makefile
Posted by TitustheTitan on Fri, 27 Jan 2017 13:52:29 GMT

Have you made a clean command yet? It's just like that, except you are calling the executable(s) you made as a result of make.

---

Subject: Re: adding "Make Test" command to makefile
Posted by hlking on Fri, 27 Jan 2017 21:10:47 GMT

This is what I have:

test:  qtest.o queue.o integer.o sll.o
    gcc qtest.o queue.o integer.o sll.o -o test

But when I run "make test", it doesn't run the executable. I have to type ./test afterwards.

I did make a clean command and it works correctly. I also tried this,

test:
    ./qtest

mirroring the clean command, but all I get from running make test is "make: test is up to date." (I changed the name of the executable that combines all my object files to qtest for this so I wouldn't have 2 different test commands).

All I can find online is really complex stuff about unit testing for makefile and things that involve files ending in .test
Oops I figured it out. The test command has to depend on the executable so it's:

test: qtest
   ./qtest

^ correct. As a general rule, never name an executable 'test'.