CSCD 437
Lab 2
Quasi Worm

SPECIFICATIONS:

Produce a self-replicating worm (running the program produces as output the original source code). The output should compile and run and produce the identical output once again.

You must use either C or Java. No other languages can be used including Python, Ruby and/or anything not C or Java!

DO NOT GOOGLE/BING/ETC. FOR ANSWERS!!! I have done so and I have those solutions in hand, including the information from the Wikipedia page. If your work is not original, you will receive 0 points.

Provide output captures that show your program self-replicates (A program that runs once and produces the output that is identical to the original is called a Quine. Run the program, compile the output, run again, then execute a diff command on the two output files to show they are the same.

Once you know the program self-replicates (produces the output of the original program one time) add the feature so your program compiles itself and runs itself. You must limit the worm to three runs.

NOTE

Your quine will be a single C or Java file. To create the worm and limit it to three runs you can have an additional C or Java file, meaning for the worm you are allowed two files of the same type (.c or .java).

EXTRA CREDIT

If you create the worm, execute the worm, and limit the worm to three runs in a single C or Java file.

HOW TO COMPLETE/SUBMIT THE LAB

You should complete this lab on your own first. Once you have completed the lab, and your teammates have completed the lab, your team should meet and determine whose code will be submitted.

WHAT TO TURN IN:

Only one team member will submit a zip file that contains a folder named your team number-worm containing

- all the source code
- output captures that proves the solution self-replicates one time (Quine)
- output captures that proves the solution is a worm
- a readme.txt file that includes
    - all member names
    - how to execute the programs (both Quine and worm)
    - discussion of any shortcomings the programs contain. If the program has no shortcoming just state 'no shortcomings'

Name your zip your team numberLab2.zip (Example: team16Lab2.zip)