2021 Fall CPSC 240-3

Final Program Test

December 14, 2021 12:00m-8:00pm

The policy is the same as in the previous tests. No need to repeat everything here.

To receive “blank credit” (20%) send me the “Hello World” or the equivalent “Happy Birthday” program.

Send the completed set of program files to holliday@fullerton.edu

Do include all source files and a bash file. Pause to be sure your are sending all the needed files. Forgetting to include a file makes the whole program inoperable.

There is no second chance to send a missing file later. I will be not be responding to email of any kind from test day until January 5. You have to submit all files on December 13.

The total points for this test is 100.

Put your name and email address on this test on two different pages. You pick the pages.

In January the final class curve for section 3 will be posted in Discord by an accomplice.

Send to holliday@fullerton.edu

Requirements

Create a program with driver file in either C or C++ and operation file in X86 assembly. That is sufficient: two source files and one bash file.

Suppose a moving object like a bicycle is moving in a straight line. How much force must be applied to the brakes to stop the vehicle?

A formula online says that the force on the brakes is 0.5 x m x v^2 / d

where	0.5 = one-half = a constant
	m = mass of bicycle (weight in kg)
	v = velocity (meters per sec)
	d = distance meters (required for complete stop)

Compute **F = 0.5 x m x v x v / d**

Make a program that inputs m, v, and d, and then computes F.

You may use any software products you can find on the web.

There is no data validation in this program.

Dialog when the program runs on an Intel microprocessor

This is Final exam by Lisa Finkelstein.

Welcome to the Lisa Braking Program.

The frequency (GHz) of the processor in machine is 2.25

Please enter the mass of moving vehicle (Kg) **3.9**

Please enter the velocity of the vehicle (meters per second): **7.1**

Please enter the distance (meters) required for a complete stop: **17.8**

[Invisible: Read cpu clock and save it]

[Invisible: Compute the braking force and save it]

[Invisible: Read the cpu clock one more time and save it.

The required braking force is **5.52244382** Newtons

The computation required 7512 tics or 3338.6 nanosec.

The main program received 3338.6 and will just keep it.

Have a nice day.

Color codes
Green: These actions are performed in the CPU, but there is no visible output.

Yellow: Output from the assembly source file.

Pink: Output from the driver.

Be sure to replace Lisa's name with your own name.

The dialog does not guarantee mathematically correct number. The dialog is a model of how to structure your output.

Dialog when the program runs on an AMD microprocessor

Some cpu’s return zero as the clock frequency. That cannot physically be right. When you call the get-frequency function if it tells you the frequency is 0.0 then ask the user for the frequency. In this case the dialog looks like the following.

Your professor may decide to test your final exam program on an AMD machine.

```
This is Final exam by Lisa Finkelstein.

Welcome to the Lisa Braking Program.

The frequency (GHz) of the processor in machine is 0.0

Please enter the mass of moving vehicle (Kg)   3.9

Please enter the velocity of the vehicle (meters per seconds):  7.1

Please enter the distance (meters) required for a complete stop:  17.8

[Invisible: Read cpu clock and save it]

[Invisible: Compute the distance and save it]

[Invisible: Read the cpu clock one more time and save it.

The required braking force is 5.52244382 Newtons

Please enter the cpu frequency (GHz):   2.25                        <== Extra statement

The computation required 7522 tics or 3338.6 nanosec.

The main program received 3338.6 and will just keep it.

Have a nice day
```

This dialog and the dialog on the previous page came from the same program executed on different platforms. In the second dialog the AMD microprocessor returns 0.0 as its frequency. That cause the program to ask the user to manually input the CPU frequency.

Partial credit is always available.

Partial credit is a good thing. A non-running program is not good.

If it appears that you can't make a program that meets all the specifications then make a program that meet some of the specifications.

For example: omit the part about reading the clock and computing run-time. Make a nice program that implements the rest of the specification.

Another for example: omit computing the braking force and only display the time (tics and nanosecs).

You are an experience programmer. Use your creative skills to implement a part of the specification. At the level of blank answer you can make a program that says welcome and your cpu speed is x.xx GHz

You do know how to do one little piece of this program. Please make my day by sending me a program representative of your great computer skills.

//End of partial credit.

Footnote: What if the user does not know the cpu of his or her own computer? Try this:
Instate the program inxi by entering the next line in the shell:

```
sudo apt install inxi
```

When that is finished enter the bash command: `inxi  -C`
with an upper case C. Just maybe the stars will shine and you'll see the frequency.

The semester has ended

I am glad you were all here this semester.   Now you have to move on to other academic studies.

I hope that you saw I am not here simply to teach one more class and collect a pay check. That went away a long time ago. I was sharing my passion with you all semester. Thanks for coming along on the journey.

Go in peace.

F. Holliday