Name:

|  |  |  |
| --- | --- | --- |
| Problem | Points | Score |
| 1 | 50 |  |
| 2 | 30 |  |
| 3 | 20 |  |
| Total | 100 |  |

Notes:

1. The first step in this exam is to create a workspace in the following directory:

/data/courses/ece\_1111/current/exams/exam\_01

Your directory should be your last name all lowercase, followed by an underscore, following by your first name (e.g, “picone\_joseph”). Set the permissions using “chmod u+rwx,g-rwx,o-rwx <lastname>” so only you have read and write permission to this directory. Create subdirectories within this directory: p01, p02, … You will use these for problems 1, 2, … respectively. Put ALL your code in these directories. Do not touch your files after the exam is over.

Failure to follow these instructions will result in a grade of 0. This preamble is part of the process of demonstrating you have basic Linux literacy.

1. Your code must be nicely formatted and well commented, or I will deduct at least 10 points per problem.
2. For this exam you are allowed to open a terminal window on your computer, you are allowed to web surf with Google, but you cannot use online chat or other interactive services.

**Problem No. 1:**

Consider the mathematical definition of a parabola:

$$x\left(t\right)=a1\*t^{2}+a2\*t+a3$$

Write a program that prints out values of this function at specific points in time. Your interface should be:

*p01.exe duration step\_size a1 a2 a3*

where $t$ starts at zero and is incremented by ($i \* step\\_size$) until $t$ exceeds the duration.

For example,

*p01.exe 10.0 1.0 1.0 2.0 3.0*

would print values of the above function for $t = 0, 1, ..., 10$.

Be careful to compute $t $using ($i \* step\\_size$) rather than "$t += step\\_size$" to avoid roundoff errors. Your loop should terminate when $t$ exceeds the duration. Be sure to read the duration and step\_size from the command line using atof().

Print values to stdout using the following format:

a1 = 100.0000

a2 = 200.0000

a3 = 20.0000

time = 0.0000 value = 0.0000

time = 1.0000 value = 275.1234

Use a make file, a header file and place your main program in a file *p01.cc*.

**Problem No. 2**: The directory tree:

*/data/courses/ece\_1111/current/resources/data/eeg\_reports/v1.0/data/000/*

contains 1000 files with an extension “\*.txt”. Each filename has a basename in format “########\_s##.txt”. Write a Unix command that counts the number of files that have the number 4 in the 5th position of the first field (e.g., “00004999\_s01.txt”) and a “9” in the last position before “s\_” (e.g., “00004999\_s01.txt” satisfies this constraint, but “00004998\_s03.txt” and “00005999\_s01.txt” do not). Your Unix command can use pipes and filters but must be one line and not create any temporary files.

Place your command along with its output in *p02/p02.txt*.

**Problem No. 3**: The Unix command “*date*” returns the current date and time. Write a C program that reads the date from stdin and prints it out. The interface should be:

*date | p03.exe*

and should display the current date (e.g., “*Wed Sep 23 08:51:17 EDT 2020*”) returned by the date command.

Place your program, which should use a make file, a header file and a main program, in */p03* so that it compiles with make.