CMSC 331

Semester Project

Proposal Due Friday 4/6, 11:59 PM, through Blackboard

Project Due Tuesday 5/15, 11:59PM, through GitHub

Updates

03/12/18: Added list of languages available on GL.

04/02/18: Raised Size Limitation to 100MB

04/30/18: Made base case more clear for problem 6

This project will give your experience working with a new language, as well evaluating that language for the appropriateness of various tasks.. All work is to be done on your own and any code should run on the GL servers. We will be using GitHub classroom to handle submission this semester. To get started, accept this homework at https://classroom.github.com/a/Jf-9Rl2h. This repository is empty, as everyone will have a different setup for this project. Don't forget to add the README.md.

You may chose any language for this project, **except for the following:** Python, C, C++, Java, Lua, Scheme, Racket, or Lisp. If you wish to use a language not on GL, that is fine, as long as it can easily be installed on GL without administrator permissions, and the installation is less than ~~25~~ 100 MB. There are many other languages installed on GL already. Your language must allow all programs to be run from a command line, with no windowing environment present.

Important information about git on GL: Git on GL is outdated, so whenever you see an

/REPO_NAME".

In addition, this will by default try to pop up a window to ask for your GitHub password. If you are on terminal, this will obviously not work, so you need to enter one of the following commands so you are prompted for your password on the terminal

```
unset SSH_ASKPASS
unsetenv SSH_ASKPASS
```

Languages on GL

The following langauges are available on GL. This list is not garunteed to be comprehensive, nor is it a list of languages that you must use or even should use for this project. Some of the languages would be very difficult to use for this assignment. There may be other languages available on GL, and many languages should be relatively easy to install in your home directory.

- bash
- csh
- fortran
- Haskell
- Matlab
- mksh
- Octave
- perl
- R
- Rhino JavaScript Interpreter
- ruby
- TCL
- tcsh
- zsh

Proposal (10 Points) Due 4/6

- The language you have chosen, including the version if it is important (ie Perl 5 vs Perl 6 is a big difference)
- Three resources you will consult when doing your project to help you with the language. These should be textbook/documentation like resources, not things like Wikipedia pages. A detailed blog, Wikibook, or Tutorial site is also ok
- Why did you chose this language?

Paper (25 Points)

Write a two page paper (double spaced, 12 point font) describing the language you chose, including a brief history of it, as well as various features of the language, according to the criteria outlined in chapters 5-9 of your textbook. You don't need to write down every possible property of the language, focus on the ones that are unique or beneficial in this language. For example if you were writing about Lua, you would definitely want to highlight multiple assignments and the features it enables, and probably its small type system.

In addition, your report should include some reflections on what about your language made implementing the problems below easy, and what made some of them difficult. You should cite any references you use, and they should be on a separate page, not included in the 2 pages.

Your paper should be submitted as PDF inside your GitHub repository. Do not email your paper to me!

README (10 points)

You must write a readme including your name, section number, and instructions on how to run your programs. You may use any language that can be installed in user space on GL, as long as the installation is less than ~~25~~ 100 MB. Your README should include all instructions needed to install the language, if needed, as well as compilation/interpretation instructions, and any specific instructions for the programs below

README Completeness (5 points)

The instructions in your README must work, and allow your code to be run with out any questions required from the instructors/graders

The following problems are designed to give you experience with a wide variety of aspects of your chosen language. Your implementations must only use what is available from a standard installation of your library, and no third party code. Each problem should be its own self contained program, titles problem#.YOUR_EXTENSION. If you were doing your assignment in Lua, your first program would be named `problem1.lua`

1. Unique Words (10 points)

For this problem, you need to write a program that takes in a file name and an integer, N, and returns the first N unique words in the file given. These can be hard coded into your program, but should be very easy to change, either through constants or as parameters to a function in your code. If the file given contains less than N unique words, all unique words in the file should be printed, without any error conditions

2. Travel Time

In this problem, you must create a program that prompts the user for a single choice, between a bicycle, car, or jet plane. Once the choice is made, you should pick a random speed for the vehicle, defined by the ranges below, and travel for "one hour" at that speed. You should continue iterating, pick a new speed each hour, until your vehicle travels 1000 miles. You should then print how many hours it took, and the maximum speed reached.

The speed ranges for the three vehicles are as follows:

- Bicycle: 5-15 miles per hour
- Car: 20-70 miles per hour
- Jet Plane: 400-600 miles per hour

Example Interaction

```
$ python3 ~/example.py
Pick a vehicle.
1.  Bicycle
2.  Car
3.  Jet Plane
Choose 1-3: 3
The number of hours it took to travel 1000 miles was 2 hours.
```

3. Sorting Words by Their Value (10 points)

For this problem, you must write a program that has a function/method/procedure that takes in an array (or equivalent data structure) of strings, and prints out the elements of that array in sorted order, one per line. You should sort the words according to a numerical value, calculated as follows: The numeric value of a word is the sum of the numeric values of the characters in it. The numeric value for a character is the ASCII value if your language encodes strings/characters using ASCII, and the Unicode Code point if your language encodes using Unicode.

4. Generating Random Numbers (10 points)

For this problem, you must create a program that when run, outputs a random number between zero and one, once every second. This should continue forever, until a user hits any key, at which point the generation of random number should stop immediately, and the following two statistics should be printed:

- The total number of random numbers generated
- The average value of the numbers generated

5. Statistics about a Matrix (10 Points)

Write a function/method/procedure that takes in a single parameter, that is a matrix, and prints the row averages and the column averages. Make sure you specify how the matrix is passed as a parameter in your language as part of your README.

6. Golomb Sequence (10 Points)

Using recursion, write a function/method/procedure in your language that computes the Golomb number, which is defined as :

G(n) = 1 + G(n - G(G(n - 1)))

The base case of this recursion is G(1) = 1

How long does your implementation take to return the value G(50)? Include this in your program so the timing is automatically printed when your program is run.

Your paper will be graded on how well you describe the language, your analysis of its use in the problems, as well as 2 points for your writing itself, including grammar, spelling, etc. and 5 points for the references.

Your README will be graded based on its clarity and ease of following

The README completeness will be graded on if the instructions described in the README works as intended

Your programs will be graded on correctness of output for given inputs. While the difference in programming languages make this difficult to generalize, it is important that the instructor/grader is able to test out your programs with different inputs, either by editing your code or calling the functions with different parameters.

Submitting Your Project

Your homework should be done on the GL system, as that is where it will be tested. Committing your code to your local repository and pushing it up to GitHub is your submission for this project. Your last submission before the due date will be taken as your submission.

Please test out your ability to commit and push your code to GitHub before the due date. If you have any problems submitting, email me or stop by office hours.