Name:

|  |  |  |
| --- | --- | --- |
| Problem | Points | Score |
| 1 | 50 |  |
| 2 | 50 |  |
| Total | 100 |  |

Notes:

1. 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.
2. You must program in Python 2.
3. You must upload your source code as one file following these formats:
   * Python: one file named lastname\_firstname\_p01.py. Must be able to run on neuronix using “python lastname\_firstname\_p01.py fname” where fname can be any valid filename.
   * C++: one file named lastname\_firstname\_p01.cc. Must be able to be compiled on neuronix using: gcc –o foo.exe lastname\_firstname\_p01.cc –lm. Must be able to be run using “foo.exe fname” where fname can be any valid filename.
   * Shell: lastname\_firstname\_p01.sh. Must be able to be run on neuronix using “sh lastname\_firstname\_p01.sh fname” where fname is any valid filename.
   * Other languages: talk to me first.
4. You must upload a second file that contains the output I have requested, and only that output. The filename must be lastname\_firstname\_p01.txt. Note the filename is all lowercase with no spaces.

**Problem No. 1**: This problem consists of two parts. Both parts should be integrated into one program that outputs two answers. You must process the file: **/data/isip/data/software\_tools/big\_files/x.tar.gz** on neuronix.

1. (50 pts) Write a program or script to compute the number of unique 7 character sequences in the file that contain 4 or more ASCII characters in the range [a-z]. For example, “zc\*a%b&” would count because it contains the letters “z”, “c”, “a” and “b”. But “###123#” would not count because there are no characters in the range [a-z] in the string. Make sure you start with the first 7 characters in the file and shift your buffer by one character each time until you reach the end of file. Test your code on a small file and make sure it works before you attempt to run the large file. Print out this information *exactly as it is shown* (replacing question marks with the actual value of course):

the number of unique 7-character strings with four or more ASCII characters is: ????

the most frequently occurring 7-character string is: ???????

1. (50 pts) Add to the previous program code that maps every non-ASCII character to space (e.g., “ ”) and then computes the sum of the decimal values of characters after mapping. Output this information *exactly as it is showN* (replacing question marks with the actual value of course):

the first 20 mapped characters are: ????????????????????

the sum of the decimal values of all characters after mapping is: ??????????

YOU MUST NOT WRITE AN INTERMEDIATE FILE for any of these computations. Everything must be done on the fly inside your program.

Generate both your answers into a text file and upload that following the instructions on the previous page. This file should have 4 lines of text containing the information requested above.

Part of the challenge here is to write your code efficiently so that it finishes in a reasonable amount of time (minutes rather than days). I will run your program on the large file and expect to see the above output. I will match that against what you included in your \*.txt file.