

Task 03 – Basic of programming

The scope of this task is to write a more “real” example of daxpy and matrix multiplication codes. You will have two problems to solve and you want to find a way to “communicate” output from one problem as a input to another.

I suggest to use your chosen compiled language to perform this task because it is harder, but you can try to use the interpreted language as well.

- 1) Create a code that given N as input generates 2 vectors, x and y, of dimension N, with values $x = 0.1$ (for all the elements) and $y = 7.1$ (for all the elements) and save them to two different file. The output filename are created in the following ways:
 - a. Read N and filename_prefix from command line (filename_prefix is a string similar to “/path/to/my/outputdir/vector_”)
 - b. Inside the code generate a string as a filename similar to this:
“/path/to/my/outputdir/vector_N<value_of_N>_x.dat” and
“/path/to/my/outputdir/vector_N<value_of_N>_y.dat” where <value_of_N> is the actual value of the input N (e.g. 10)
- 2) Create a second code that, using a configuration file, take as input the filename of the file with x, the filename of the file y, the value of N and the scalar value $a = 3$, and a prefix_output (similar to before). Then calculate the usual

$$\vec{d} = a\vec{x} + \vec{y}$$

One example of configuration file in c: <https://github.com/welljsjs/Config-Parser-C>
The output is a file containing the vector d.

- 3) Are you able to compile the two codes with a single makefile? [examples of makefile for c: <https://www.cs.colby.edu/maxwell/courses/tutorials/maketutor/>]
- 4) Working with text files or binary files (https://en.wikipedia.org/wiki/Binary_file) rise some problems. In the scientific computing community (and in HPC in particular) the HDF5 file format is often use to read and write large data structures [https://en.wikipedia.org/wiki/Hierarchical_Data_Format]. Try to modify your codes to generate and read information from HDF5 files.
- 5) Scientific computing often rely on external libraries. One of the most common (because is free and it is written in c) is the GNU Scientific Library (GSL). Can you rewrite your codes using the vector sum from this library?

<https://www.gnu.org/software/gsl/doc/html/vectors.html>

You will need to use in particular this function **gsl_vector_axpby**