Getting started with MPI

1   Installing MPI

Instructions on Linux (with root privileges)

```
$ apt-get install mpich
```

Instructions on Linux (without root privileges)

1. Get and untar the archive at:

   http://www.mpich.org/static/downloads/3.4.1/mpich-3.4.1.tar.gz

2. Make a `mpich-install` directory, outside `mpich-3.4.1`, where MPI executables will be installed.

3. Create an environment variable containing the path to your MPI installation directory:
   `$ export MPIHOME=<homedir>/.../mpich-install`
   Add this command to your ∼/.bashrc file so you do not have to invoke it each time you open a new session.

4. Prepare the Makefiles to build the MPI executables:
   `$ ./configure --disable-fortran --disable-f77 --disable-fc --prefix=$MPIHOME`

5. Build the library: `$ make`

6. Install the library: `$ make install`

7. Include the path to the binaries to the environment variable `PATH`:
   `$ export PATH=$MPIHOME/bin:$PATH`
   Add this line to your ∼/.bashrc file

8. `mpicc` and `mpirun` commands should now be available.

Instructions for MacOS

```
brew install mpich
```

Instruction for Windows

Follow instructions at (did not try ourselves):
https://docs.microsoft.com/en-us/message-passing-interface/microsoft-mpi?redirectedfrom=MSDN.

2 Mnemonics and π

A short mnemonic for remembering the first seven decimal digits of π is "How I wish I could calculate pi". Much longer texts have however been conceived for π, which have the particular property to correspond to its decimal digits.

For every word in the text, in the order they appear, we keep its size as a digit of π.

- If the size of a word is less than 10, add the corresponding digit

- If the size of a word is equal or greater than 10, then the two digits need to be added to the sequence.

- Blanks, points and new line are ignored

- Other special characters (such as "!", ":" and "(") contribute as the "0" digit

- A number is copied as is in the digit sequence

Exercise A

Download the template sequential code `text2digits.c` and the π text `pi-text.txt` available in the Teams directory.

Complete the missing parts in order to perform the conversion of texts in digit sequences by following the rules above. Compile your code with `gcc` and execute your program by considering the texts that you can find at the same address and verify whether it is able to correctly generate the first digits of π.

Exercise B

Write an MPI parallel version of your program. The text can be read by all processors, while the conversion of the text needs to be split and performed in parallel: every processor can work on its own portion of the text. Pay particular attention to not cut words when splitting up the text. At the end of the computations, the processors do not need to communicate, they can just print their partial results on the screen, in an arbitrary order. Compile now with `mpicc` and execute with `mpirun -np <nb_proc> ./<exec_name>`.

Exercise C

You may have noticed that the random access to the standard input by the processors does not allow us to have a clear visualization of the digit sequence. Instead of printing the digits as soon as they are generated, each process can keep them in an array of integers, and print them all at the end of the computations. By using the function `MPI_Barrier`, we can force the processors to print the content of their local arrays in the right order.