

# CS Basics - Exercises

## Basics of the C language

C. Grothoff and E. Benoist

Fall Term 2017-18

### 1 Hello World

**Exercise 1** *Download, compile and run GNU Hello.*

```
$ wget http://ftp.gnu.org/gnu/hello/hello-2.10.tar.gz
$ tar xvf hello-2.10.tar.gz
$ cd hello-2.10
$ ./configure --prefix=$HOME
$ make
$ make install
$ ~/bin/hello
```

### 2 Test your code!

**Exercise 2** `$ make check` # in hello-2.10 directory

### 3 Go International

Enable multiple locales on your system (or all), and then switch your primary language:

```
Exercise 3 # dpkg-reconfigure locales
$ export LANGUAGE="de_CH:de"
$ ~/bin/hello
$ export LANGUAGE="fr_CH:fr"
$ ~/bin/hello
$ export LANGUAGE="it_CH:it"
$ ~/bin/hello
```

## 4 Read the documentation!

```
Exercise 4 $ man hello
$ info hello
$ cd doc
$ makeinfo --pdf hello.texi
```

## 5 Simplify!

**Exercise 5** *Create a minimal “Hello World” (in a directory `hello/`):*

```
#include <stdio.h>
int
main ()
{
    printf ("Hello world!\n");
    return 0;
}
```

*Create a built-system in a rush using the following `Makefile.am`:*

```
bin_PROGRAMS = \
    hello
hello_SOURCES = \
    hello.c

# apt-get install automake autoconf
$ cd hello/
$ autoscan
$ mv configure.scan configure.ac
```

*Edit `configure.ac`, adding `AM_INIT_AUTOMAKE` under “Checks for programs” and editing the `AC_INIT` line, replacing the arguments with the correct program information.*

```
$ touch NEWS README AUTHORS ChangeLog
$ autoreconf -fi
$ ./configure --prefix=$HOM$
$ make
$ ./hello
```

## 6 Write a first C program

**Exercise 6** *Write a program (based on the examples) that computes the area and perimeter given the radius of a circle, the radius is given as a number you enter on stdin.*

*You should define a constant `PI`. And use the `stdio.h`*

*Update your makefile for compiling your C program.*

## 7 Write a program for reading a file on stdin

**Exercise 7** Write a program that takes as input on standard input. Reads char after char and transforms all lowercase letters into uppercase letters. (You need to use a *while* loop and have some *if* statement, the syntax is the same as in Java).

## 8 Bubble Sort

**Exercise 8** Write a program that asks the user for a string and sorts the letters in ascending (respective to the ASCII code of each letter, i.e. its value) order using Bubble Sort. You can see a demonstration of Bubble sort in the following link.

For the students knowing already C, this exercise is much too easy, they should implement "quick sort" (see algorithm description provided by Wikipedia : <https://en.wikipedia.org/wiki/Quicksort>).

Example of use of Bubble Sort and Quick Sort: <http://www.benoist.ch/coursWebProgramming/examples/base/sort.php>

## 9 Gettextize (advanced, optional)

**Exercise 9** Install and run *gettextize* (follow the instructions carefully) and use `gettext("Hello world!\n")` to create a minimal internationalized version of your "Hello world" program.