Einführung in die Programmierung für Physiker

WS 2019/2020 – Marc Wagner

Alessandro Sciarra: sciarra@itp.uni-frankfurt.de

Exercise sheet 13

To be corrected in tutorials in the week from 03.02 to 07.02.2020

Exercise 1 [*File I/O*]

Reading and writing information from and to external files is a very important aspect of any programming language. In this exercise we will practice this feature in `C`.

A typical use case is to read from file some input information at the beginning of our program, to use it in some calculation to produce some results and to store these into an output file. In our case, the goal is to calculate some partial sums

$$S_N = \sum_{i=1}^{N} \frac{1}{i^p}$$

for some integer values of $p > 1$ and some equidistant values of N in the interval $[N_0, N_1]$. We will read from a configuration file the values of p, N_0, N_1 and of δN, which will fix which partial sums have to be calculated. At the end, our program will create a file containing, on each line, the values of N and S_N.

(i) Use e.g. the `bash` command `echo "3 2 1002 10" > input.dat` to create a possible input file.

(ii) Write a `C` program to read the `input.dat` file and to produce the output file containing the partial sums as well as the information about how many terms of the series were summed up. Its name, e.g. for $p = 3$, should be `3-series_partial_sums.dat`.

  (a) Implement a `void readInputFile(/*...*/)` function to read the input file.

  (b) Implement a `double getPartialSum(/*...*/)` function to calculate a fixed S_N.

  (c) Implement a `void writeOutputFile(/*...*/)` function to write the output file.

Here, some additional remarks.

- You are free to structure your data as you wish, but a new data type to store the input information could be appropriate and handy to be passed to functions.
- What do the functions `fopen`, `fscanf`, `fprintf` and `fclose` do? Which should be used where? Did you test the `FILE`-pointers for `NULL` values?
- How can you pass the data to be written to file to the `writeOutputFile` function? What about using a couple of arrays? How do you reserve memory for them?
- How would you declare the output filename? Can you use the value of p to build up the filename that contains any p value given in input? Do you know the `sprintf` function of the `stdio.h` library?

Exercise 2 [*Exam like assignment*]

Consider the following piece of code.

```
int a[6] = {2, 3, 5, 7, 11, 13};
int *b = &(a[3]);
int c[2][2] = { { *(b-3), a[0] },
                {  *b-3,  a[5] } };
```

For each of the expressions below,

- specify whether the syntax is allowed (i.e. whether it is accepted by the compiler);
- comment on whether the expression does not lead to the probably desired[1] result;
- state its type, e.g. `int`, `double*`, etc.;
- predict, if possible, what is printed out if the expression is used in a `printf` statement with the corresponding appropriate specifier.

(i) `a[7]`

(ii) `&(a[2])`

(iii) `a+b`

(iv) `b[-1]`

(v) `b+2`

(vi) `*b`

(vii) `sizeof a`

(viii) `c[1]`

(ix) `*(*(c+1)+1)`

(x) `(float)(c[0][0]/c[1][0])`

[1] Do not worry: At the exam you will not be asked to read our mind. . . Here the question is meant to discuss further the meaning of the code.