Einführung in die Programmierung für Physiker

WS 2017/2018 – Marc Wagner

Francesca Cuteri: cuteri@th.physik.uni-frankfurt.de
Alessandro Sciarra: sciarra@th.physik.uni-frankfurt.de

Exercise sheet 13

To be corrected in tutorials in the week from from 05/02/2018 to 09/02/2018

Exercise 1 [*File I/O*]

Reading and writing information from and to external files is one of the most important aspects 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 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 `3-series_PS.dat` file containing the partial sums as well as the information about how many terms of the series were summed up.

   (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 questions*]

(i) What operators can you use to access members of some structure? Assume that the pointer `myTrianglePtr` has been declared to point to `struct Triangle` having a member `height` of type `float`. Assume also that the address of the structure `myTriangle` has been assigned to `myTrianglePtr`. Rewrite the expression `myTrianglePtr->height` with an equivalent one using the structure member operator `.` to access the relevant member.

(ii) Write a declaration statement for a variable of some defined structure type. Using which keyword can you create an alias for a derived data type and how? Rewrite your declaration statement making use of your alias.

(iii) State whether the following are *true* or *false* and, in the latter case, explain why.

(a) The structure definition

```
struct Triangle {
    float base;
    float height;
};
```

reserves space in memory for the struct being defined.

(b) Structure members must be variables of built-in data types.

(c) One cannot just declare variables of a given structure type, but also arrays whose elements are of a given structure type and pointers to structure type variables.

(d) Structure can be compared using the comparison operators `==` and `!=` and can be operands of arithmetic operations.

(iv) Write a function that takes a number of seconds, convert it into hours, minutes and seconds and print the result to the screen.

(v) Write a recursive function that, given two integers a and $b > 0$, returns a^b.