UFAZ - Bachelor of Computer Science

System Programming

PW02 : advanced C programming

For each exercice, we expect the student to write a program, compile it and run it without errors of several examples. Test sets and comments are as important as the code itself.

Exercice 1

Write a program which copies a file `foo` into a new file (which you need to create) `bar`, using the functions `fopen`, `putc`, `getc` and `fclose`.

Exercice 2

The `Unix` command `tail` displays, by default, the 10 last lines of a text file. The option `-n` allows to change this number of lines. Write a program which performs the same job as `tail`.

To analyse the arguments, we suggest you use the function `getopt`. To find the n last lines of the files, we suggest to read the m last bytes of the file and to count the number of occurrences of `\n` starting from the end. If this number is not sufficient, we start over. When the correct number of `\n` is found, the program displays all the characters until the end of the file.

Exercice 3

We aim at writing a program which reads several floating-point numbers, stores them in an array (we assume there is no more that 100 values), sorts this array and then displays it on the standard output.

We propose to define 3 functions:

1. `int read_array (double *a, int max_elem, int *nb_elem)` which receives as argument an array `a` with at most `max_elem`qui recoit un tableau en parametre contenant au plus `max_elem` elements, lit les valeurs et les range dans le tableau, puis renvoie le nombre delements lus dans `nb_elem`. La valeur de retour de cette fonction doit etre 0 si la lecture sest bien passee, -1 sinon.
2. `void sort_array (double *a, int nb_elem)` which sorts the `nb_elem` elements of the array `a`.
3. `void display_array (double *a, int nb_elem)` which displays the `nb_elem` elements of the array `a`.

Write the code of the functions as well as the `main` function.

Exercice 4

Write a program to compute the perimeter and the area of a circle from its radius. The program should be structured in several files : `main.c` (which contains the `main` function), `perimeter.c` (which contains the `perimeter` function), `area.c` (which contains the `area` function) and `pi.h` (which contains an approximate value of Π.

1. Write all the programs, and provide the smallest compilation line to create an executable program.
2. Decompose this compilation command to have a single command for each file, and then a link-editing command. If you change something in the file `area.c`, give the minimum commands required to recompile the whole program. What happens if you modifiy `pi.h` instead.
3. Write a makefile, which shall automate the compilation process. Add a new target which removes all intermediate files as well as the final runnable program.