Alarms.

a) Write a program in ANSI C/POSIX which:

- Reads from stdin a (possibly very long) sequence of alarm times in the form of positive decimal floating point numbers separated by white spaces. The numbers refer to times from the moment when the program has been started, in fractions of a second.

- Triggers the alarms, writing to stdout a sequence of the sum of positive differences between the desired and actual triggered alarm times measured so far (in fractions of a second, as decimal floats separated by '\n'). No alarm may be triggered sooner than its time specifies. Implement the triggering of alarms so that the the writing to stdout can be easily switched off (so that only the final sum can be reported).

- Terminates when the standard input stream ends (EOF) and all the alarms have been triggered.

The objective is to minimise the (sum of) absolute reported differences, i.e. to trigger the alarms as precisely as possible.

b) Propose a benchmark which demonstrates that your implementation is better in comparison with other implementations (e.g. those discussed in the lecture). Explain your choice of the benchmark.

c) Run your program on the benchmark, evaluate your program quantitatively, explain your measurements, make a nice graph etc.

Hand out a ZIP file hw2.zip containing all the necessary files, and a report hw2.pdf.