

Programming Competition

In this competition, you are supposed to implement a *pattern matcher*. You may implement either the algorithm presented in class or any other algorithm you find. The rules are simple and the prizes are high.

- The pattern matcher takes two inputs – a text file, `text.txt`, and a string, `s`, (as a command line argument).
- It outputs the number of occurrences of `s` in `text.txt` as well as the prefix `text.txt` ending with the first occurrence of `s`.
- *Ignore* new lines (`\N`) and carriage returns (`\R`) (in contrast to `grep`).
- Your pattern matcher may be written in C, C++, Java, or an ML dialect (OCaml, SML/NJ, MoscowML, ...). No other languages allowed.
- You *must not* use any libraries other than facilities to input/output files. In particular, do not use any string manipulation packages/libraries. Don't use code found on the Internet.
- Your solutions will be checked for cheating. It is up to the jury to decide what cheating is.
- Submission deadline: **December 12!** Send an email to Jörg Kreiker containing:
 - the source code of your solution
 - instructions on how to compile and use your implementation (should be compilable on any standard Linux/Windows platform)
 - a description of the algorithm you have implemented and the optimizations made
 - the answers to the challenges below (if any) and the running times of your tool in order to answer these questions
- You may work in groups of up to three people.

Challenges

Your pattern matcher will be evaluated according to the size of the input it can handle and the time it takes to do so. At the following URL you can download a compressed archive containing four benchmark text files and a description of the challenges.

<http://www7.in.tum.de/um/courses/auto/ws0809/benchmark.zip>.

The archive contains `faust.txt`: Goethe's *Faust*, size app. 200KB; `homer.txt`: Homer's *Iliad* and *Odyssey* in English, size app 1.4MB; `chromosome22.txt`: the complete primary structure of the human chromosome number 22, a string composed of the letters A, C, G, T, and N, size app. 34MB; `123.txt`: a file containing a single string of 1s, 2s, and 3s, size app. 42MB; and `challenge.txt`: a description of the challenges.

Prizes

- We will verify your solutions in a *live session* during the tutorial on *December 18* and determine the winner.
- The winner is the group whose tool solves most challenges. If more than one group solves all challenges, running times are considered.
- The winner will receive *a bottle of champagne* and *a crate of beer*. The runner-up will receive champagne only.
- There will be a special *jury's prize* awarded to the most elegant solution.

Good luck!