



Programming Test

1. Objective

Implement a 'readline' function with a prototype similar to GNU readline: `char *readline (const char *prompt);`
The function will display a prompt and wait for user input.

1.1 Minimal Features

Append characters in a buffer, return buffer on reception of carriage return or end-of-file.

1.2 Additional Features

1. Handle backspace key
2. Handle left/right/home/end keys to move cursor and allow insertion of characters in the middle of the buffer
4. Handle delete key
5. Handle up and down keys to implement a history feature
6. Any extra feature you can think of :)

2. Requirements

- Written in C language (C99 variant)
- Compliant with Linux kernel coding style (at least for naming and indentation)
- No external dependencies except for POSIX libc functions
- Only use POSIX ``getchar`` and ``putchar`` to read/write to TTY
- For additional features, assume the terminal is compatible with VT100 escape sequences

3. Expected Delivery

A complete project to build a demonstration Linux program (sources + Makefile or build script).
The project will be delivered as an archive containing a git repository.

Optional:

The repository is expected to have a clean history:

- each additional feature should be in a separate self-contained commit
- at each commit, the project should compile cleanly and the demonstration program should be functional

4. Evaluation Criteria

The project will be evaluated based on several criteria:

- it needs to be functional
- it needs to build cleanly (without warnings, using a recent gcc and ``-Wall`` build option)
- coding style and code readability are important
- code needs to be commented where it is useful
- code robustness is a must: special attention must be taken to avoid buffer overflows and any unwanted side effects

