Smart\_calc

1. Create a GitHub repository.
2. Create a Dockerfile and built the image based on Ubuntu. Setup the following: gcc, check.h, lcov, pkg-config.
3. Create a CMake from Makefile.
4. Make tests using CMake by cases which are considered in the initial programming code (folder src/test/).
5. Create a git submodule for external library and create a shell script to download it to the initial project (prereqs.sh).
6. Create a shell script which automatically creates Makefile (using CMake), builds the project and launches the initial code (build.sh).
7. Create a shell script which creates tests using CMake, launches it and export the html file (using lcov) with tests coverage (test.sh).
8. Create a GitHub-workflow for prereqs.sh, build.sh, test.sh which launches these scripts..



1. Create a documentation on quick guide and description.

Structure

The project represents itself a smart calculator. Command line is used for input and output. There is possible to make calculations both using numbers and elementary mathematical functions. If something wrong with the input, tests provide an analysis of parsing. Besides tests analyze if the calculation or polish notation is correct or not.

Legend:

Yellow: backend, initial programming code.

Calc.c - make calculation

Parser.c - parse input

to\_polish.c - convert input to polish notation

Stack.h - additional file

Pink: all the tests, using framework «check.h».

Blue: entry.c - main function, launches the code (stdin/stdout).

Green: makefile

Turquoise: readme file (general description).