

Problem Set #3: Some Problems to Solve

Goals

- practice algorithmization
- work with standard input and output

!Warning

This assignment must be submitted by **04/03/2022 23:59:59** . Use the slack channel for discussion [#problems-to-solve](#).

What should be done?

As the name suggests, the goal of this assignment is to solve several problems. Specifically, these are the following problems:

- [Mighty Method](#)
- [Perfect Isolation](#)
- [Racer](#)
- [Famous Cheese Factory](#)

Handing over the project

The assignment is submitted via the [Git](#) version management system on the server <http://git.kpi.fei.tuke.sk/> . You will submit the solution to this task as part of your project.

The structure of your project will look like this:

```
├── ps3/
│   ├── cheesefactory.c
│   ├── mm.c
│   ├── isolation.c
│   └── racer.c
└── README
```

The names of the files correspond to the names of the individual problems to be solved.

- **README**- A file that will indicate the designation of your group that you attend for exercises in the form:

```
GROUP : A1
```

!Warning

It is important that your files maintain the structure listed. If any of the files are in the repository, but in a different folder, it will be considered an error and such a project will not be considered correct! If, on the other hand, there are extra files or folders in your project, these will not be considered an error.

!Warning

Folder, file and file content names are **README case** -sensitive!

Evaluation and testing

Your evaluation will depend on the results of the tests that your assignment successfully passes. The following will be verified:

- The structure of your project (whether it contains all the necessary files).
- Static analysis of your code using the **cppcheck**.
- Checking for memory leaks with a tool **valgrind**
- The presence of global variables in your code.
- The functionality of your implementation.

The solution of each assignment must have a function **main()** in order for your assignment to run. You can find the file name for a specific assignment on the text page of that assignment.

This type of assignment does not test functions that you create in the program! This means that you can create as many functions as you want, but they must remain in the solution file.

The input is tested against the output your program prints to the screen against the input your program reads from standard input. Sample inputs and outputs against which you can test your assignments are found with each assignment.

Your code will be compiled by the gcc compiler with the following switches and libraries:

```
$ gcc -std=c11 -Werror -Wall -lm
```

Testing of your solutions will be done automatically every 3 hours.

Your solutions will once again pass the originality check. Therefore, when working on your assignment, behave according to the rules of the [code of ethics](#) ! If you submit an assignment that is not yours, you will be expelled from the course!

Programming

The course covers advanced topics in the *C language and Arduino* microcontroller programming .



Except, where otherwise noted, content on this site is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International \(CC BY-NC-SA 4.0\)](#).