**Test task**

**Task:** Develop a scripting language interpreter for testing shared functions libraries. A program should be developed with C language usage as a console application for the OS GNU/Linux (x86 processor architecture) with API POSIX.1-2001 usage.

**Scripting language syntax:** The script file is a text file with an arbitrary name. The rule applies: one

line is one command.

There are two commands in the language.

1. Loading the library

*use <so\_name>*

*<so\_name>* - the path to the file \*.so

Possible errors:

* the file not found;
* the file is not a shared library.

1. Library function call

*call <func\_name>*

*<func\_name>* - the name of the function

Possible errors:

* the function is not found;
* the library is not loaded.

Syntax errors must be accompanied by interpreter warnings and must not cause the script to stop executing. In case of command execution error – execution the script is terminated.

Additionally, you can implement support for comments ('**#**', '**;**') and line breaks ('**\**') and **interactive mode of the application** (entering commands directly from the terminal).

**Description of test libraries:**

All test libraries should be developed with C language usage. All exported functions should have prototype:

*void example (void);*

**Requirements to the application:**

The application should be launched from the command line with only one parameter – the path to the file script. An example:

*sotest test.sc*

**Example of the script:**

*use test.so*

*call example*

*call test1\_func*

*use other.so*

*call func1*

**The result of the test task:**

* Application source code (compilation instruction should be included)
* Self-testing result of the application (to demonstrate that the application works properly in candidate’s own machine), including:
  + Screenshot/console output of the application execution
  + The script ( .sc file ) used for testing
  + The source of the libraries used for testing