two programs. Programs communicate to each other via sockets.

**Program 1.**

Program has two threads and one common buffer.

Thread 1. Reads a string from the user input. Checks that string contains digits only and one’s length limits 64. After check the string must be sorted by descending and all even numbers must be replaced to “KB” letters. Then the string places into common buffer and the thread is waiting for the next user input.

Thread 2. Processes the data placed into the common buffer. After thread have gotten the data buffer is cleared. Thread prints the received data on the screen and calculates sum of all numeric elements in the received data. Calculated is sent to program 2. Further thread 2 is waiting for next data.

Notes for program 1:

1. Threads communication are synchronized. Thread 2 does not constantly check the common buffer.
2. Both programs are independent on each other states. It means that sudden stop of program 2 wont affect the user input in program 1. If program 2 was restarted, then reconnection will happen.

**Program 2.**

Is waiting data from program 1. When data is received program calculates data length. If the data length is larger than 2 symbols and the length multiple of 32 then program shows the message about received data, otherwise program shows the error message. Further program is waiting for the next data.

Notes for program 2:

1. Both programs must be independent on each other states. It means that sudden stop of program 1 must not raise the problem in program 2. If the connection was lost, the program must waits for new connection.

* Each program is in its own folder.
* Build process does not require any special configuration.
* Sources must be compiled using **cmake or make in Linux**. The source folder is free of junk: unused source or resource files, intermediate build files, and so on.