

Guide to compile and run your practicals

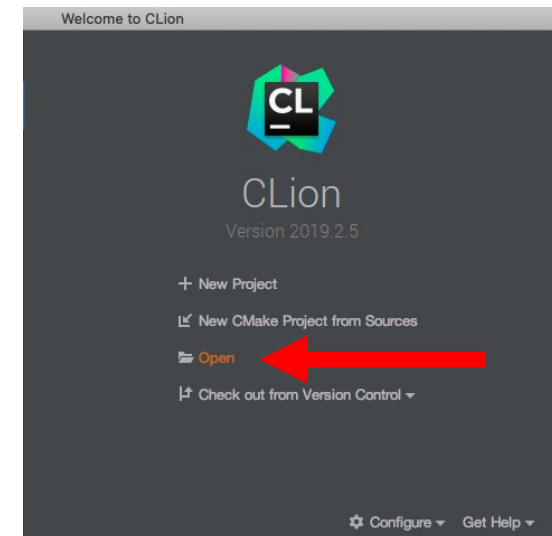
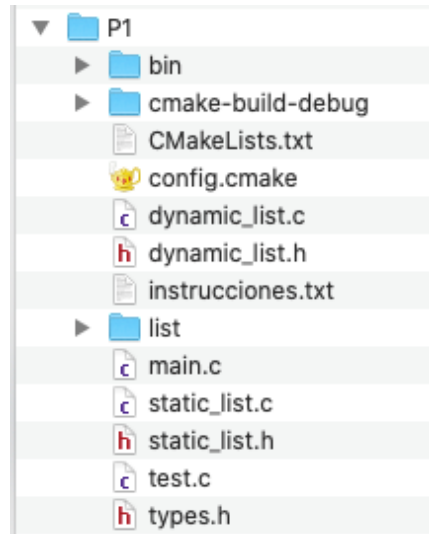
Programming II

Faculty of Computer Science



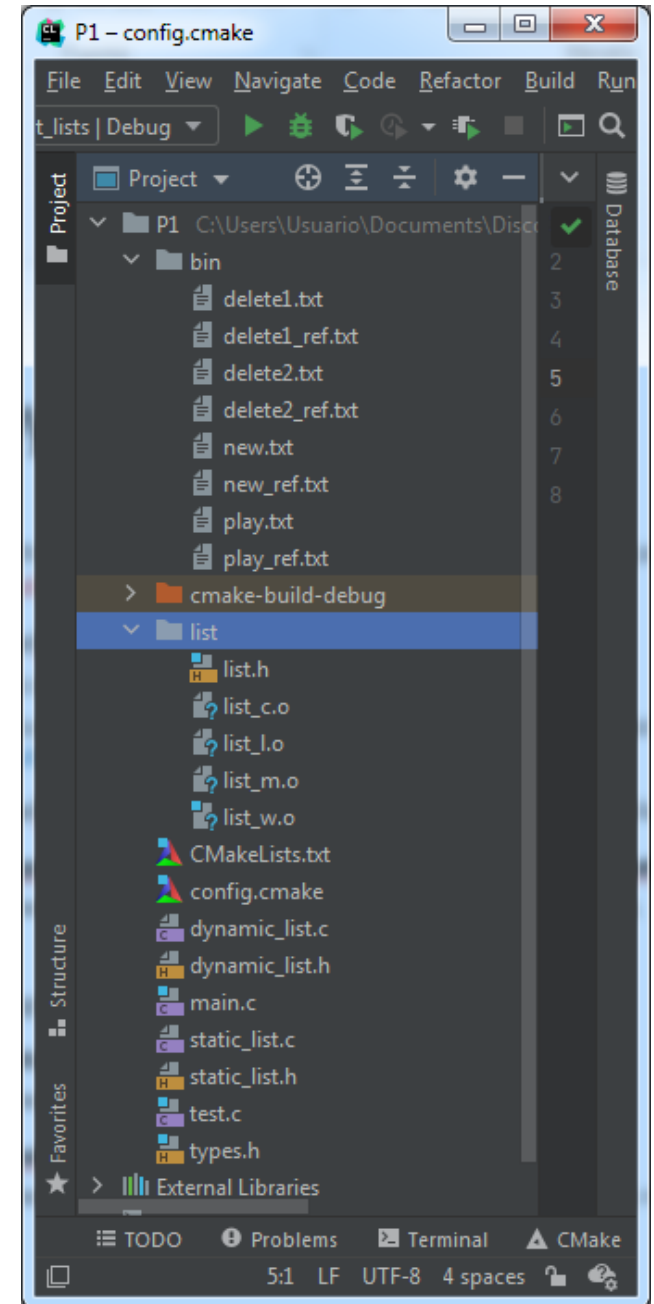
Template project

1. The first step will be to download and decompress the ZIP file available at Moodle that contains all the files necessary for the practical.
2. Next, from CLion, open (*Open*) the folder containing the files of the practical.
 - This folder already includes the definition of a CLion project **ready to compile**.



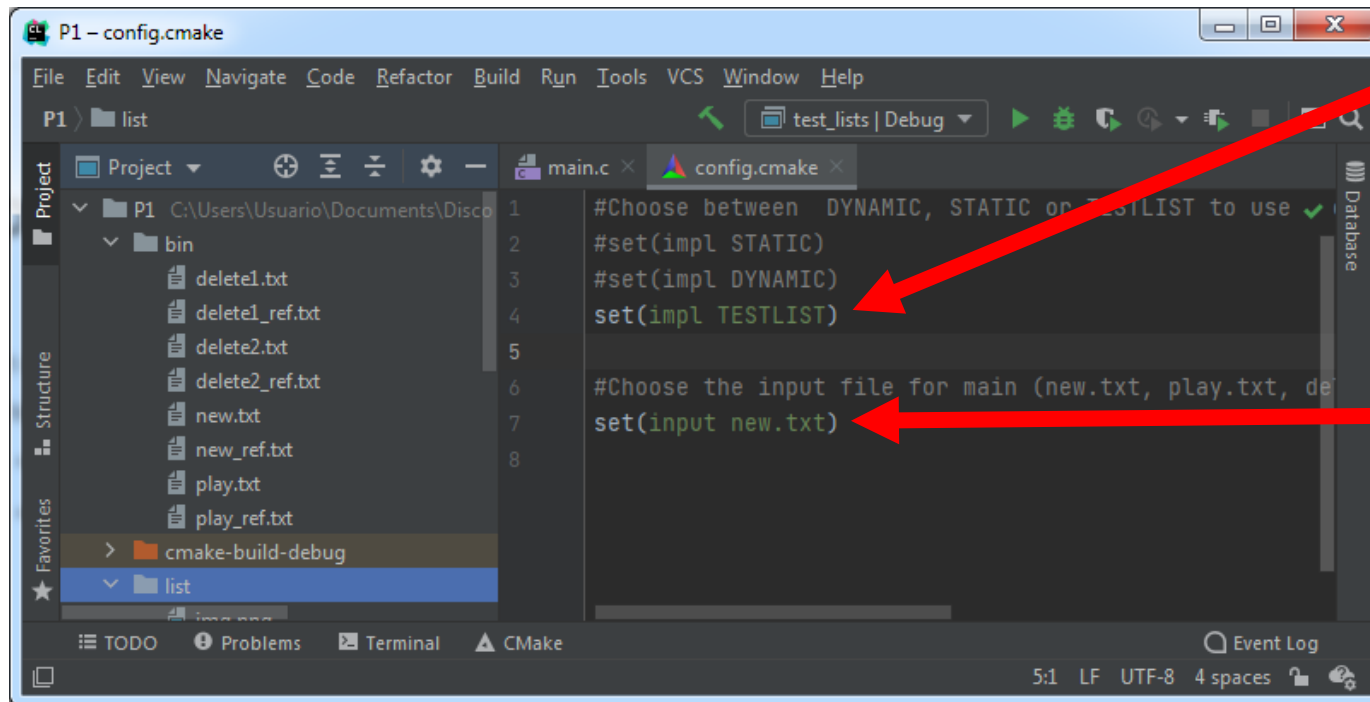
Template project

- The project includes all the C source files necessary to carry out the practical.
The student must not add any other .c or .h file.
- **Folder bin:**
 - Where the executables are generated when compiling.
 - Contains the input test files necessary to test the operations of the practical (.txt files) and the result of its execution (*_ref.txt files).
- **Folder list:** contains (compiled for Windows, Linux and MacOS) the library of the ADT list to be used at the beginning of the practical.
- **main.c, types.h, dynamic_list.h, dynamic_list.c, static_list.h, static_list.c**
 - Source files to be filled in to complete the practical.
- **test.c:** Program for testing an ADT List.
- **CMakeLists.txt:** Configuration file for CMake. **It must not be modified.**
- **config.cmake:** File to be modified when testing the program. It must be made according to the instructions included.



Configuring compilation and execution

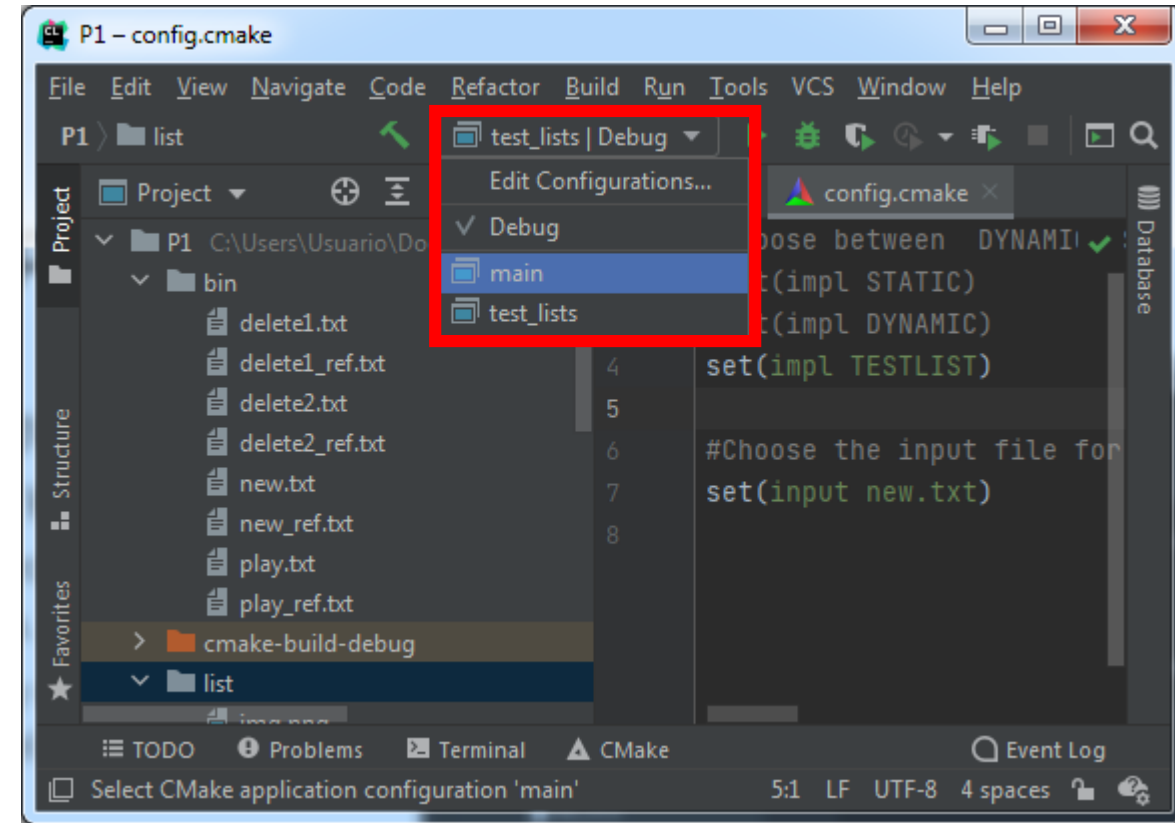
- Go to file **config.cmake** and select the following options (by commenting or uncommenting the corresponding lines):



- Version of the ADT List implementation to be used to compile and run the program.
- Input file for the program (to be chosen among the ***.txt** files in folder **bin**).

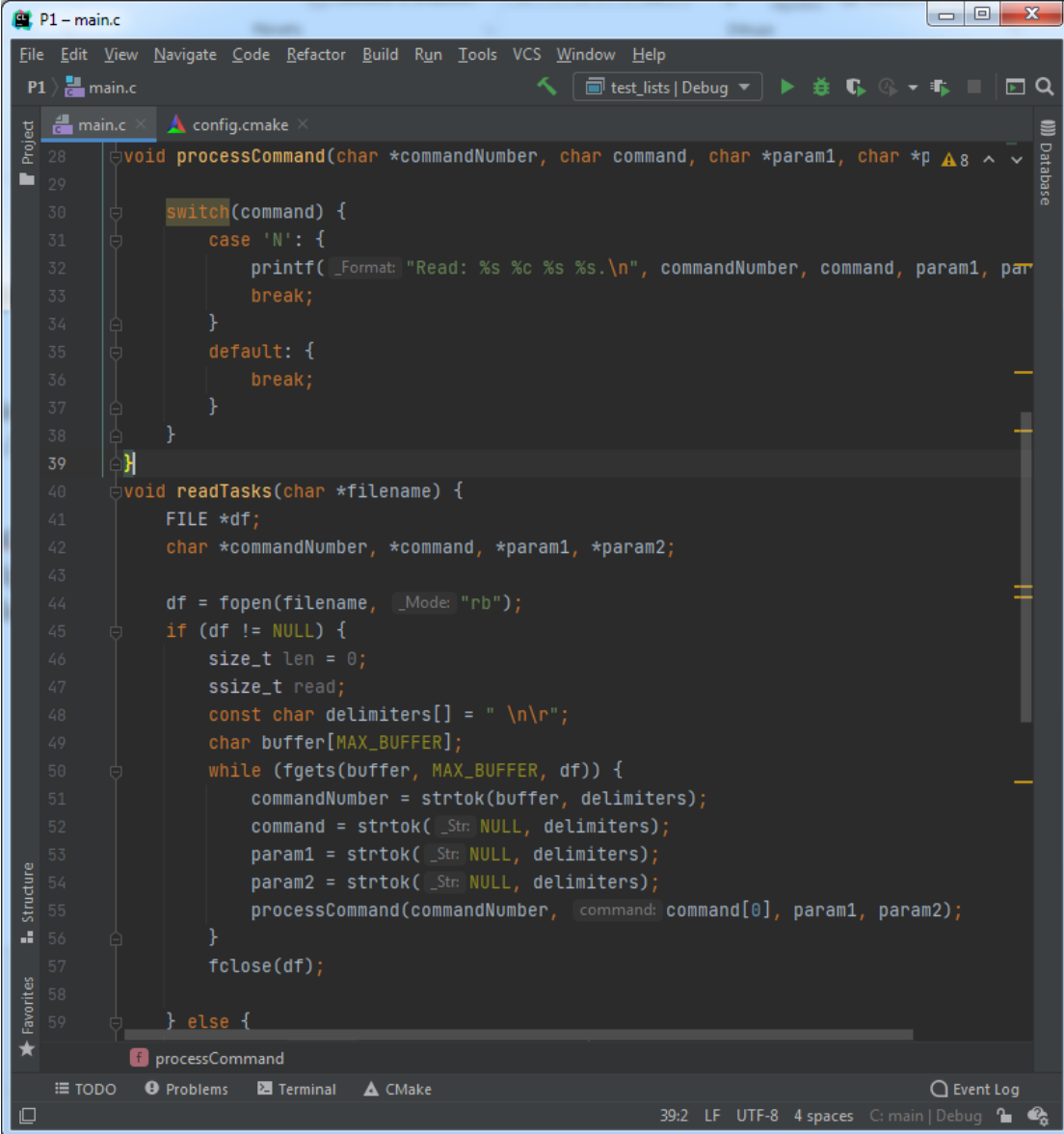
Configuring compilation and execution

- The template project is already configured with two compilation and run *targets*:
 - **main**: to compile and run the main program of the practical.
 - **test_lists**: to compile and run the testing program that checks the proper functioning of the ADT List implementations.
- You must select the corresponding one at the CLion drop-down menu according to which one you want to **compile** or **run**.



Programming the practical

- Complete the C source code of the **.c** and **.h** files included with the project.
- The file **main.c** contains the basis source code of the main program, including the code necessary for reading the input files.



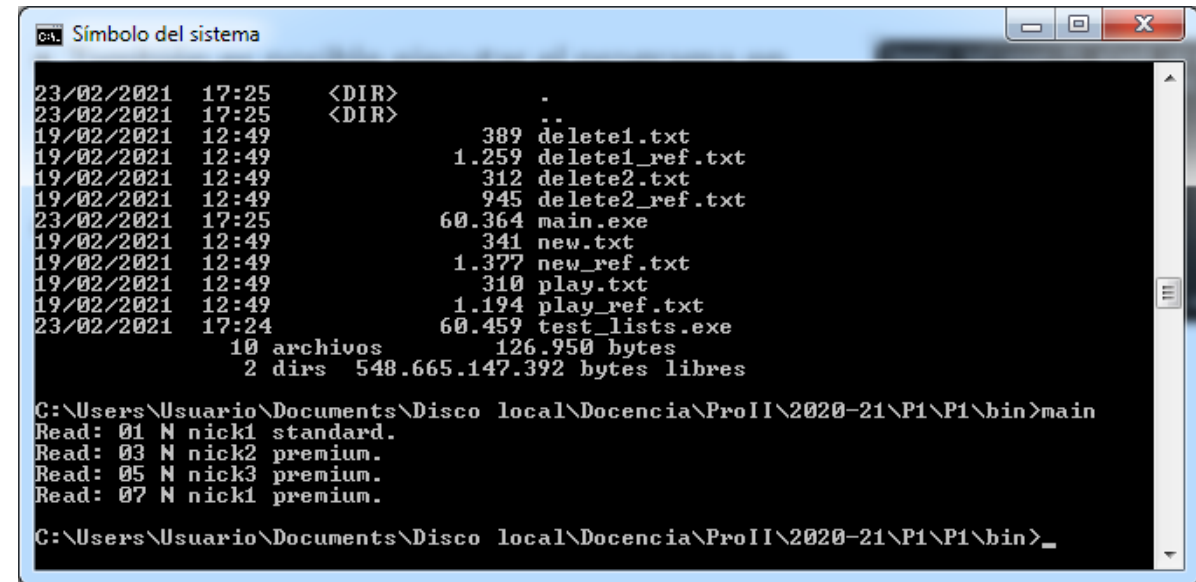
The screenshot shows a C code editor with the following code in `main.c`:

```
28 void processCommand(char *commandNumber, char command, char *param1, char *p
29
30     switch(command) {
31     case 'N': {
32         printf(_Format: "Read: %s %c %s %s.\n", commandNumber, command, param1, par
33         break;
34     }
35     default: {
36         break;
37     }
38 }
39
40 void readTasks(char *filename) {
41     FILE *df;
42     char *commandNumber, *command, *param1, *param2;
43
44     df = fopen(filename, _Mode: "rb");
45     if (df != NULL) {
46         size_t len = 0;
47         ssize_t read;
48         const char delimiters[] = " \n\r";
49         char buffer[MAX_BUFFER];
50         while (fgets(buffer, MAX_BUFFER, df)) {
51             commandNumber = strtok(buffer, delimiters);
52             command = strtok(_Str: NULL, delimiters);
53             param1 = strtok(_Str: NULL, delimiters);
54             param2 = strtok(_Str: NULL, delimiters);
55             processCommand(commandNumber, command: command[0], param1, param2);
56         }
57         fclose(df);
58     } else {
59
60     }
```

The editor interface includes a menu bar (File, Edit, View, Navigate, Code, Refactor, Build, Run, Tools, VCS, Window, Help), a toolbar with icons for running, debugging, and other actions, and a sidebar with Project, Structure, and Favorites views. The status bar at the bottom shows the current line (39:2), encoding (LF), character set (UTF-8), indentation (4 spaces), and the active file (C: main | Debug).

Running from the console

- It is also possible to run the program from the console using the compiled binary files (generated at folder **bin**).



```
23/02/2021 17:25 <DIR> .
23/02/2021 17:25 <DIR> ..
19/02/2021 12:49      389 delete1.txt
19/02/2021 12:49    1.259 delete1_ref.txt
19/02/2021 12:49      312 delete2.txt
19/02/2021 12:49      945 delete2_ref.txt
23/02/2021 17:25    60.364 main.exe
19/02/2021 12:49      341 new.txt
19/02/2021 12:49    1.377 new_ref.txt
19/02/2021 12:49      310 play.txt
19/02/2021 12:49    1.194 play_ref.txt
23/02/2021 17:24    60.459 test_lists.exe
                10 archivos      126.950 bytes
                2 dirs  548.665.147.392 bytes libres

C:\Users\Usuario\Documents\Disco local\Docencia\ProII\2020-21\P1\P1\bin>main
Read: 01 N nick1 standard.
Read: 03 N nick2 premium.
Read: 05 N nick3 premium.
Read: 07 N nick1 premium.

C:\Users\Usuario\Documents\Disco local\Docencia\ProII\2020-21\P1\P1\bin>_
```

- Along with the documentation of the practical, a **script** for the automatic testing of the program is provided. This script will have to be run from the console of the **reference server**.