FACULTY OF COMPUTER SCIENCE
Programming II

Running the Script (P2)

Together with the documentation of the practical, a zipped file `script.zip` is provided. Once decompressed, it generates a folder named `script` that contains a script file (`script.sh`) and two subfolders (`script_test` and `script_minimos`). These files allow us to test the list and the stack (by using `test_product_list.c` and `test_bid_stack.c`) and the main program (`main.c`). To do the latter, the two subfolders above contain all the input files needed joint with their corresponding expected outputs (files ending in `_ref.txt`).

In order to run this test script correctly, you must do the following. Firstly, create a directory in the reference server. Then, copy into that directory:

1. All the content of the folder `script` obtained when decompressing `script.zip`; that is, the script file `script.sh`, the subfolders `script_test` and `script_minimos` and the files `test_product_list.c` and `test_bid_stack.c`.
2. All the `.c` and `.h` files with the code of our practical (`main.c`, `types.h`, `product_list.c`, `bid_stack.c`, `product_list.h` and `bid_stack.h`).

Next, from the terminal and after going to that directory, we must give the file `script.sh` permissions for execution using:

```
chmod u+x script.sh
```

Finally, run said script. Two options are available:

1. `./script.sh` (equivalent to `./script.sh -p main`)
2. `./script.sh -p test`

In the first case, the main program (`main.c`) is executed with all the input files contained in the folder `script_minimos`. As a result, the files containing the outputs returned by the main program (`new_output.txt` for `new.txt`, `bid_output.txt` for `bid.txt` and so on) are created in that same directory. In the case of the second option, the test programs of the list and the stack are executed (`test_product_list.c` and `test_bid_stack.c`), and the outputs obtained are compared with the expected ones (files ending in `_ref.txt`).

The script includes an option `-v` that, apart from indicating whether the output is correct or not, also shows on screen those lines where the program's output differs from the reference output:

```
./script.sh -v
```

Partial and final deliveries

- For the first partial delivery to be assessed with PASS, the result of executing:

  ```
  ./script -p test
  ```

  must be:

  ```
  Tests global result (checkpoint #1 - April 8th): OK
  ```

- For the second partial delivery to be assessed with PASS, the result of executing:

  ```
  ./script.sh -p main
  ```

  must be `OK` for the input files `new.txt` and `bid.txt`:

  ```
  Checkpoint #2 result (April 22th): OK
  ```

- For the final delivery to be assessable with PASS, the result of executing:`./script.sh -p main`
  must be `OK` for the six input files and, therefore, `Main global result` must be `OK` too.