Source code to verify the results from the paper titled
"Rapidly Verifiable XMSS Signatures"
by Joppe W. Bos and Andreas Hülsing and Joost Renes and Christine van Vredendaal
The paper is online at: https://eprint.iacr.org/2020/898

This is the modified RFC code which includes the counter generation in the *signature generation*.
Compile with:

```
gcc -D SHIFT=10 -D LEN=3 -g -O3 -Wextra -Wpedantic -o test xmss_tests.c params.c randombytes.c xmss_core_fast.c hash.c hash_address.c wots.c utils.c xmss_commons.c fips202.c xmss.c sha2.c
```

and set the SHIFT parameter to run the counter up to 2^SHIFT.

In xmss_tests.c one can change the message length which is by default
#define XMSS_MLEN (32+8)
--> 32 bytes for the message and 8 bytes for the counter.

There are various macros once can enable / disable in params.h

* Set the macro ORIG to 1 if you want to execute the original RFC code.
  - The macro PRECOMP can be set to 1 to use the hash precomputation trick as described in:
      Cryptology ePrint Archive: Report 2020/470
      LMS vs XMSS: Comparison of Stateful Hash-Based Signature Schemes on ARM Cortex-M4
      Fabio Campos and Tim Kohlstadt and Steffen Reith and Marc Stoettinger
    Set the PRECOMP macro to 0 to not use this technique.

* Set the macro ORIG to 0 if you want to use the algorithms described in the paper
  related to signature generation and verification with a counter.
  - The macro PRECOMP can be set to 1 to use the hash precomputation trick as described in:
      Cryptology ePrint Archive: Report 2020/470
      LMS vs XMSS: Comparison of Stateful Hash-Based Signature Schemes on ARM Cortex-M4
      Fabio Campos and Tim Kohlstadt and Steffen Reith and Marc Stoettinger
    Set the PRECOMP macro to 0 to not use this technique.
  - By default the signature generation and verification are computed by the implementation.
    If you only want to run the verification (for instance on an embedded device) do the following:
       1. Set the macro PRINT_SIGN to 1.
       2. Compile the code.
       3. Run and generate the signatures into sign.h:
          ./test 2> sign.h
       4. Set PRINT_SIGN back to 0 and set VERIFY_ONLY to 1.
       5. Compile the code.

6. Run the code (this now only computes the verification using the counters).

License