Quad-Double computation package
Copyright (C) 2003-2018
==========

Revised 30 Oct 2018

To build this library, follow the steps below.
Some system specific notes are at the end of this file.

Build Instructions
==========

  1. Run the configure script by typing

        ./configure

     The script will attempt to automatically detect various system-dependent
     variables used during compilation (such as the C++/fortran compiler,
     compiler flags, and linker flags).

     If you want to specify a particular C++ / F90 compiler and their flags,
     you can set them as environmental variables. For example:

       FC=ifc FCFLAGS="-O2 -FR" ./configure

     Important variables are

       CXX   C++ compiler to use
       CXXFLAGS C++ compiler flags to use
       CC   C compiler to use (for C demo program)
       CFLAGS   C compiler flags to use (for C demo program)
       FC   Fortran 90 compiler
       FCFLAGS   Fortran 90 compiler flags to use
       FCLIBS   Fortran 90 libraries needed to to link with C++ code.

     See ./configure --help to see other options.

  3. The configure script should also have created the files 'config.h' and
     'include/qd/qd_config.h', which will contain the compile time
     defines. Examine these and edit them if necessary. In most cases
     no edits are necessary, since the options are detected when configure
     was run.

  4. Type "make". This will build the library, and necessary Fortran

wrappers.

5. Optionally, one can build and run some simple test programs.
   To do this, type "make check". Some programs run during this
   phase is a good demonstration of how to use the qd library in C++.

6. You can now install the QD library by issuing "make install".

7. If you want to build some sample programs written in C++
   you can type "make cpp-demo".

8. If you want to build some sample programs written in Fortran 90,
   you can type "make fortran-demo".

9. If you want to compile the Experimental Mathematician's Toolkit,
   type "make toolkit". This will compile the Fortran-90 codes in
   the toolkit cirectory, including the "mathinit" and "mathtool"
   Read the "README" file in the toolkit directory for additional details.

System-Specific Notes

Linux with Intel processors

You can use g++ to compile the C++ code, which is a part of all
Linux or other Unix distributions. The Fortran 90 codes
can be compiled using the gfortran compiler available at

   https://gcc.gnu.org/wiki/GFortranBinaries

Alternatively, one can use the Intel compilers, available at:

   http://www.intel.com/software/products/compilers/clin/
   http://www.intel.com/software/products/compilers/flin/

One can specify specific compilers for the configure script, as in:

   ./configure CXX=g++ FC=gfortran

Apple (OS X)

For Apple OS X Intel-based systems, it is recommended that you use
the g++ compiler and related command-line tools, available via this
URL:

```
   https://developer.apple.com/downloads/index.action

See "Command-line tools" for your version of OS X. The above URL
requires
a registered Apple ID. It may be necessary to install Apple's Xcode
package first.

The gfortran compiler for Mac OS X can be downloaded from:
   https://gcc.gnu.org/wiki/GFortranBinaries

After installing these compilers, in the main qd directory type

   ./configure CXX=g++ FC=gfortran FCFLAGS=-m64

then type "make" to construct the library.  See the "README" file on
how to construct a compile-link script for your own codes.

IBM (Power)
-----------

With IBM's xlC/xlf90 compilers, you may want to experiment with
--enable-fma option which uses a faster code but relies on the
compiler to generate a fused multiply-accumulate instruction.
WARNING: since the compiler is not required to produce such
instructions, this is not guaranteed to work.  Please test before
using.
```