User Manual:

****

PROGRAM : Enumeration Mathematical Library[EML]
Project Manager : Mosfiqur Rahman (mr986@drexel.edu)
Last Modified : 3rd June, 2016.

****

PROGRAM PURPOSE:

The name of our project is "Enumeration Mathematical Library" acronym as EML. This is a numerical Math library for C and C++ programmers. Firstly, we have started with Number Theory, and subsequently, will explore other areas of mathematics. Our project is an open source one, which will be published under the GNU General Public License. Our intention is to follow the open source business model. So, it will be really useful for the scientific computing community. As it is developed in an open source environment, the development cost is minimal, and our main profit will be through donations and supporting corporations.

Current Coverages:

Complete Coverages:

- ✔ Arbitrary large numbers
- ✔ Basic Number theoretic Functions
  For Example:
  - ■ Mod[k,n]
  - ■ Quotient[m,n]
  - ■ Divisibility[m,n]
  - ■ GCD[n1, n2, ……]
  - ■ LCM[n1, n2, ……]
  - ■ Integer Digit[n,b]
- ✔ Matrix Algebra
- ✔ P-adic fields
  For Example:
  - ■ Quote Notation
  - ■ Local-global principle
- ✔ Common Combinatorial functions
  For Example:
  - ■ Multinomial[n1, n2, ……]
  - ■ Hyperfactorial[n]
  - ■ Sub-factorial[n]
  - ■ Fibonacci

Substantial Coverages:

- ✔ Arbitrary Precision Number
  - ■ Precision is needed to be defined
- ✔ Complex numbers
  - ■ Equity
  - ■ Ordering
- ✔ Power Series

Basic Usable Areas:

- Scientific computing and mathematical research.
- Cryptography and security systems.
- Complex Engineering mathematical problems with an extremely fast calculating experience.

User Interface:

External Interface:

- EML is a C++/C library that can be accessed through its relevant header files.
- As an example, for C, to include arbitrary large number library from EML, #include <eml/arbl.h>. Then while compiling, use the proper flags for EML like -leml in GCC. For C++, you can #include <eml/arb>. Then while compiling, use the proper flags for EML like -leml in GCC.

Data Interface & Graphical Rendering:

- EML is generally stand-alone and it won't require any data interface in general cases. However, in case of cluster computing, data interface is required for data transfer between different nodes. Here, OpenMP and MPI libraries will handle it, which is still under testing phase.
- In general, graphical renders aren't fast and resource consuming. As current graphics packages and such libraries are not fast enough, these will be re-written.

Current Features:

- It depends on Glibc C Standard Library but it's strictly implemented in ANSI C format, for portability.
- For each part of the library[refer to system manual], there's a single header file, which contains all function definitions, type definitions, structure definitions and constants.
- Our provided libraries are only compatible with Linux operating system and the architecture support is limited to x86-64 processor types.
- It's compiled as 64-bit and 32-bit binaries for some specific operating systems such as Ubuntu or other Debian based OS.

Further Supports:

- ✓ Static Library:
  - A .a file that has no dependencies but cannot be shared and can be used for development purposes that choose to distribute this library. Library of object code which is linked with, and becomes part of the application.
- ✓ Dynamic Library:
  - A .so file or dynamic library that is dynamically linked at run time but statically aware. The libraries must be available during compile/link phase. The shared objects are not included into the executable component but are tied to the execution.
- ✓ Source Code:
  - We are delivering the source code as a .zip or tar archive. The source can be used to modify the library and build upon different platforms.