LCL

Generated by Doxygen 1.8.14

Contents

Chapter 1

Main Page

Hello, world!

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

Chapter 5

Namespace Documentation

5.1 LCL_Bool Namespace Reference

Functions

- int IntToBoolVec (bool *out, int I, int m=-1)
- int BoolVecToInt (const bool *x, int len)
- int BitSize (int I)
- int Inner (bool *x1, bool *x2, int len)
- int Weight (bool *x, int len)
- void BitwiseAnd (bool *x1, bool *x2, bool *out, int len)
- void BitwiseXor (bool *x1, bool *x2, bool *out, int len)
- void copy (bool *src, bool *dst, int len)
- void print (bool *x, int len, const char *pre=NULL)
- bool increment (bool *x, int len)
- void zeros (bool *x, int len)
- bool nextUniquePerm (bool *out, const bool *in, int len)
- void fromString (bool *out, const char *in_s)
- int ReedDecoder (bool *x, int R, int M, bool *c=NULL, bool *e=NULL)

5.1.1 Function Documentation

5.1.1.1 BitSize()

```
int LCL_Bool::BitSize (
            int I )
```

Definition at line 35 of file LCL_Bool.cpp.

5.1.1.2 BitwiseAnd()

```
void LCL_Bool::BitwiseAnd (
            bool * x1,
            bool * x2,
            bool * out,
            int len )
```

Definition at line 57 of file LCL_Bool.cpp.

5.1.1.3 BitwiseXor()

```
void LCL_Bool::BitwiseXor (
            bool * x1,
            bool * x2,
            bool * out,
            int len )
```

Definition at line 63 of file LCL_Bool.cpp.

5.1.1.4 BoolVecToInt()

```
int LCL_Bool::BoolVecToInt (
            const bool * x,
            int len )
```

Definition at line 88 of file LCL_Bool.cpp.

5.1.1.5 copy()

```
void LCL_Bool::copy (
            bool * src,
            bool * dst,
            int len )
```

Definition at line 69 of file LCL_Bool.cpp.

5.1.1.6 fromString()

```
void LCL_Bool::fromString (
            bool * out,
            const char * in_s )
```

Definition at line 155 of file LCL_Bool.cpp.

5.1.1.7 increment()

```
bool LCL_Bool::increment (
            bool * x,
            int len )
```

Definition at line 75 of file LCL_Bool.cpp.

5.1.1.8 Inner()

```
int LCL_Bool::Inner (
            bool * x1,
            bool * x2,
            int len )
```

Definition at line 43 of file LCL_Bool.cpp.

5.1.1.9 IntToBoolVec()

```
int LCL_Bool::IntToBoolVec (
            bool * out,
            int I,
            int m = -1 )
```

Definition at line 10 of file LCL_Bool.cpp.

5.1.1.10 nextUniquePerm()

```
bool LCL_Bool::nextUniquePerm (
            bool * out,
            const bool * in,
            int len )
```

Definition at line 102 of file LCL_Bool.cpp.

5.1.1.11 print()

```
void LCL_Bool::print (
            bool * x,
            int len,
            const char * pre = NULL )
```

Definition at line 51 of file LCL_Bool.cpp.

5.1.1.12 ReedDecoder()

```
int LCL_Bool::ReedDecoder (
            bool * x,
            int R,
            int M,
            bool * c = NULL,
            bool * e = NULL )
```

Definition at line 166 of file LCL_Bool.cpp.

5.1.1.13 Weight()

```
int LCL_Bool::Weight (
            bool * x,
            int len )
```

Definition at line 145 of file LCL_Bool.cpp.

5.1.1.14 zeros()

```
void LCL_Bool::zeros (
            bool * x,
            int len )
```

Definition at line 96 of file LCL_Bool.cpp.

5.2 LCL_ConsoleIn Namespace Reference

Functions

- int GetCommandOutput (char *dest, int n, const char *comm)

  Retrieves the standard-out of a system command and puts it in a character array.

5.2.1 Function Documentation

5.2.1.1 GetCommandOutput()

```
int LCL_ConsoleIn::GetCommandOutput (
            char * dest,
            int n,
            const char * comm )
```

Retrieves the standard-out of a system command and puts it in a character array.

Parameters

dest	- Destination c-string.
n	- Size of c-string.
comm	- The command to be passed to the system terminal.

Returns

0 if successful, non-zero otherwise.

Remarks: This function makes use of popen, pclose etc. This requires including <stdio.h> and compiler options -std=c++11 -U__STRICT_ANSI__

Definition at line 8 of file LCL_ConsoleIn.cpp.

5.3 LCL_ConsoleOut Namespace Reference

Functions

- ostream & LOut ()
- void dout ()
- void warning (const char *message, const char *function_name=NULL, const char *class_name=NULL)
- void error (const char *message, const char *function_name=NULL, const char *class_name=NULL)
- void comment (const char *message, const char *function_name=NULL, const char *class_name=NULL)
- double secs (clock_t tic, clock_t toc)

Variables

- int LOut_Pad = 0

5.3.1 Function Documentation

5.3.1.1 comment()

```
void LCL_ConsoleOut::comment (
            const char * message,
            const char * function_name = NULL,
            const char * class_name = NULL )
```

Definition at line 68 of file LCL_ConsoleOut.cpp.

5.3.1.2 dout()

```
void LCL_ConsoleOut::dout ( )
```

Definition at line 19 of file LCL_ConsoleOut.cpp.

5.3.1.3 error()

```
void LCL_ConsoleOut::error (
            const char * message,
            const char * function_name = NULL,
            const char * class_name = NULL )
```

Definition at line 50 of file LCL_ConsoleOut.cpp.

5.3.1.4 LOut()

```
ostream & LCL_ConsoleOut::LOut ( )
```

Definition at line 12 of file LCL_ConsoleOut.cpp.

5.3.1.5 secs()

```
double LCL_ConsoleOut::secs (
            clock_t tic,
            clock_t toc )
```

Definition at line 24 of file LCL_ConsoleOut.cpp.

5.3.1.6 warning()

```
void LCL_ConsoleOut::warning (
            const char * message,
            const char * function_name = NULL,
            const char * class_name = NULL )
```

Definition at line 32 of file LCL_ConsoleOut.cpp.

5.3.2 Variable Documentation

5.3.2.1 LOut_Pad

```
int LCL_ConsoleOut::LOut_Pad = 0
```

Definition at line 9 of file LCL_ConsoleOut.cpp.

5.4 LCL_Int Namespace Reference

A collection of useful functions for integer arrays.

Functions

- void sort (int *x, int n, bool desc=true, int *a=NULL, int method=0)
  Sorts int vector x of length n in descending (ascending) order.
- int randi (int in_min, int in_max)
- void randi (int *x, int n, int in_min, int in_max)
- void print (int *x, int n, const char *pre=NULL)
- void copy (int *dst, const int *src, int n)
- void sub (int *dst, const int *src, int n, int m, int i0=0)
  Copies a sub-array of length m < n to a new array.
- void concat (int *top, const int *bottom, int n, int m)
  Concatenates vector top of length n with bottom of length bottom, the result of which is stored in top.
- void randperm (int *x, int n, int x0=0)
  Randomly permutes the input array x. Optionally adds a constant x0 to each element.

5.4.1 Detailed Description

A collection of useful functions for integer arrays.

5.4.2 Function Documentation

5.4.2.1 concat()

```
void LCL_Int::concat (
            int * top,
            const int * bottom,
            int n,
            int m )
```

Concatenates vector top of length n with bottom of length bottom, the result of which is stored in top.

Definition at line 74 of file LCL_Int.cpp.

5.4.2.2 copy()

```
void LCL_Int::copy (
            int * dst,
            const int * src,
            int n )
```

Definition at line 62 of file LCL_Int.cpp.

5.4.2.3 print()

```
void LCL_Int::print (
            int * x,
            int n,
            const char * pre = NULL )
```

Definition at line 51 of file LCL_Int.cpp.

5.4.2.4 randi() [1/2]

```
int LCL_Int::randi (
            int in_min,
            int in_max )
```

Definition at line 39 of file LCL_Int.cpp.

5.4.2.5 randi() [2/2]

```
void LCL_Int::randi (
            int * x,
            int n,
            int in_min,
            int in_max )
```

Definition at line 45 of file LCL_Int.cpp.

5.4.2.6 randperm()

```
void LCL_Int::randperm (
            int * x,
            int n,
            int x0 = 0 )
```

Randomly permutes the input array x. Optionally adds a constant x0 to each element.

Definition at line 80 of file LCL_Int.cpp.

5.4.2.7 sort()

```
void LCL_Int::sort (
            int * x,
            int n,
            bool desc = true,
            int * a = NULL,
            int method = 0 )
```

Sorts int vector x of length n in descending (ascending) order.

Parameters

x	- Vector to be sorted.
n	- length of x
desc	- sorts in descending order if true, ascending otherwise
a	- stores the original index of x[i] at position a[i]
method	- The sorting algorithm to be used. 0 - Bubble sort.

Definition at line 9 of file LCL_Int.cpp.

5.4.2.8 sub()

```
void LCL_Int::sub (
            int * dst,
            const int * src,
            int n,
            int m,
            int i0 = 0 )
```

Copies a sub-array of length m $<$ n to a new array.

Parameters

dst	- Destination array.
src	- Source array.

Definition at line 68 of file LCL_Int.cpp.

5.5 LCL_Mat_GF2 Namespace Reference

Functions

- bool ∗∗ construct (int n, int m)
- void destruct (bool ∗∗A, int n, int m)
- void copy (bool ∗∗A, int n, int m, bool ∗∗O)

- void print (bool **A, int n, int m, char *pre=NULL, bool header=true, ostream &inOS=cout)
- void add (bool **A, bool **B, int n, int m, bool **O)
- void times (bool **A, bool **B, int n, int m, int p, bool **O)
- void transpose (bool **A, int n, int m, bool **O)
- void addrow (bool **A, int n, int m, int i_t, int i_s)
- void swaprow (bool **A, int n, int m, int i_1, int i_2)
- void addcol (bool **A, int n, int m, int j_t, int j_s)
- void swapcol (bool **A, int n, int m, int j_1, int j_2)
- void rowechelon (bool **A, int n, int m)
- bool ** nullspace (bool **A, int n, int m, int &d)
- void eye (bool **A, int n, int m)
- void zeros (bool **A, int n, int m)
- void random (bool **A, int n, int m)

5.5.1 Function Documentation

5.5.1.1 add()

```
void LCL_Mat_GF2::add (
              bool ** A,
              bool ** B,
              int n,
              int m,
              bool ** O )
```

Definition at line 56 of file LCL_Mat_GF2.cpp.

5.5.1.2 addcol()

```
void LCL_Mat_GF2::addcol (
              bool ** A,
              int n,
              int m,
              int j_t,
              int j_s )
```

Definition at line 98 of file LCL_Mat_GF2.cpp.

5.5.1.3 addrow()

```
void LCL_Mat_GF2::addrow (
              bool ** A,
              int n,
              int m,
              int i_t,
              int i_s )
```

Definition at line 84 of file LCL_Mat_GF2.cpp.

5.5.1.4 construct()

```
bool ** LCL_Mat_GF2::construct (
            int n,
            int m )
```

Definition at line 10 of file LCL_Mat_GF2.cpp.

5.5.1.5 copy()

```
void LCL_Mat_GF2::copy (
            bool ** A,
            int n,
            int m,
            bool ** O )
```

Definition at line 35 of file LCL_Mat_GF2.cpp.

5.5.1.6 destruct()

```
void LCL_Mat_GF2::destruct (
            bool ** A,
            int n,
            int m )
```

Definition at line 22 of file LCL_Mat_GF2.cpp.

5.5.1.7 eye()

```
void LCL_Mat_GF2::eye (
            bool ** A,
            int n,
            int m )
```

Definition at line 223 of file LCL_Mat_GF2.cpp.

5.5.1.8 nullspace()

```
bool ** LCL_Mat_GF2::nullspace (
            bool ** A,
            int n,
            int m,
            int & d )
```

Definition at line 139 of file LCL_Mat_GF2.cpp.

5.5.1.9 print()

```
void LCL_Mat_GF2::print (
            bool ** A,
            int n,
            int m,
            char * pre = NULL,
            bool header = true,
            ostream & inOS = cout )
```

Definition at line 43 of file LCL_Mat_GF2.cpp.

5.5.1.10 random()

```
void LCL_Mat_GF2::random (
            bool ** A,
            int n,
            int m )
```

Definition at line 239 of file LCL_Mat_GF2.cpp.

5.5.1.11 rowechelon()

```
void LCL_Mat_GF2::rowechelon (
            bool ** A,
            int n,
            int m )
```

Definition at line 112 of file LCL_Mat_GF2.cpp.

5.5.1.12 swapcol()

```
void LCL_Mat_GF2::swapcol (
            bool ** A,
            int n,
            int m,
            int j_1,
            int j_2 )
```

Definition at line 104 of file LCL_Mat_GF2.cpp.

5.5.1.13 swaprow()

```
void LCL_Mat_GF2::swaprow (
            bool ** A,
            int n,
            int m,
            int i_1,
            int i_2 )
```

Definition at line 90 of file LCL_Mat_GF2.cpp.

5.5.1.14 times()

```
void LCL_Mat_GF2::times (
            bool ** A,
            bool ** B,
            int n,
            int m,
            int p,
            bool ** O )
```

Definition at line 64 of file LCL_Mat_GF2.cpp.

5.5.1.15 transpose()

```
void LCL_Mat_GF2::transpose (
            bool ** A,
            int n,
            int m,
            bool ** O )
```

Definition at line 76 of file LCL_Mat_GF2.cpp.

5.5.1.16 zeros()

```
void LCL_Mat_GF2::zeros (
            bool ** A,
            int n,
            int m )
```

Definition at line 231 of file LCL_Mat_GF2.cpp.

5.6 LCL_Maths Namespace Reference

Functions

- unsigned long long int fact (int n)
- unsigned long long int nCr (int n, int r)

5.6.1 Function Documentation

5.6.1.1 fact()

```
unsigned long long int LCL_Maths::fact (
            int n )
```

Definition at line 8 of file LCL_Maths.cpp.

5.6.1.2 nCr()

```
unsigned long long int LCL_Maths::nCr (
            int n,
            int r )
```

Definition at line 16 of file LCL_Maths.cpp.

5.7 LCL_MenuUtils Namespace Reference

Functions

- void getOneInt (int &inOut, istream &inIS, ostream &inOS, const char *inMessage)
- void getOneDouble (double &inOut, istream &inIS, ostream &inOS, const char *inMessage)
- void getOneChar (char &inOut, istream &inIS, ostream &inOS, const char *inMessage)
- void getOneCString (char *inOut, istream &inIS, ostream &inOS, const char *inMessage)
- void getOneString (string &inOut, istream &inIS, ostream &inOS, const char *inMessage)

5.7.1 Function Documentation

5.7.1.1 getOneChar()

```
void LCL_MenuUtils::getOneChar (
            char & inOut,
            istream & inIS,
            ostream & inOS,
            const char * inMessage )
```

Definition at line 27 of file LCL_MenuUtils.cpp.

5.7.1.2 getOneCString()

```
void LCL_MenuUtils::getOneCString (
            char * inOut,
            istream & inIS,
            ostream & inOS,
            const char * inMessage )
```

Definition at line 36 of file LCL_MenuUtils.cpp.

5.7.1.3 getOneDouble()

```
void LCL_MenuUtils::getOneDouble (
            double & inOut,
            istream & inIS,
            ostream & inOS,
            const char * inMessage )
```

Definition at line 18 of file LCL_MenuUtils.cpp.

5.7.1.4 getOneInt()

```
void LCL_MenuUtils::getOneInt (
            int & inOut,
            istream & inIS,
            ostream & inOS,
            const char * inMessage )
```

Definition at line 9 of file LCL_MenuUtils.cpp.

5.7.1.5 getOneString()

```
void LCL_MenuUtils::getOneString (
            string & inOut,
            istream & inIS,
            ostream & inOS,
            const char * inMessage )
```

Definition at line 45 of file LCL_MenuUtils.cpp.

5.8 LCL_Utils Namespace Reference

Contains handy functions.

Functions

- double rand_d ()

  Returns a random double between 0 and 1.
- int rand_i (int min, int max)

  Returns a random int between min and max.
- int rand_i (int num)

  Returns a random int between 0 and num.
- int flipBitN (int index, int Nbit)

  Flips the Nth bit of binary expansion of index.
- int kDelta (int i, int j)

  Returns 1 if and only if i==j.
- bool Bn (int index, int bit)

  Returns the Nth bit of binary expansion of index.
- void getComment (istream &inStr)

  Displays a commented out line from an input stream if it begins with one.
- int GCD (int inA, int inB)

  (I think) GCD calculates the Greatest Common Divisor between inA and inB.
- int factorize (int inC, int *inFactors)

  Factorizes an integer inC and places the factors in an array inFactors.

5.8.1 Detailed Description

Contains handy functions.

5.8.2 Function Documentation

5.8.2.1 Bn()

```
bool LCL_Utils::Bn (
            int index,
            int bit )
```

Returns the Nth bit of binary expansion of index.

Definition at line 38 of file LCL_Utils.cpp.

5.8.2.2 factorize()

```
int LCL_Utils::factorize (
            int inC,
            int * inFactors )
```

Factorizes an integer inC and places the factors in an array inFactors.

Definition at line 74 of file LCL_Utils.cpp.

5.8.2.3 flipBitN()

```
int LCL_Utils::flipBitN (
            int index,
            int Nbit )
```

Flips the Nth bit of binary expansion of index.

Definition at line 25 of file LCL_Utils.cpp.

5.8.2.4 GCD()

```
int LCL_Utils::GCD (
            int inA,
            int inB )
```

(I think) GCD calculates the Greatest Common Divisor between inA and inB.

Definition at line 54 of file LCL_Utils.cpp.

5.8.2.5 getComment()

```
void LCL_Utils::getComment (
            istream & inStr )
```

Displays a commented out line from an input stream if it begins with one.

Definition at line 44 of file LCL_Utils.cpp.

5.8.2.6 kDelta()

```
int LCL_Utils::kDelta (
            int i,
            int j )
```

Returns 1 if and only if i==j.

Definition at line 34 of file LCL_Utils.cpp.

5.8.2.7 rand_d()

```
double LCL_Utils::rand_d ( )
```

Returns a random double between 0 and 1.

Definition at line 10 of file LCL_Utils.cpp.

5.8.2.8 rand_i() [1/2]

```
int LCL_Utils::rand_i (
            int min,
            int max )
```

Returns a random int between min and max.

Definition at line 14 of file LCL_Utils.cpp.

5.8.2.9 rand_i() [2/2]

```
int LCL_Utils::rand_i (
            int num )
```

Returns a random int between 0 and num.

Definition at line 20 of file LCL_Utils.cpp.

Chapter 6

Class Documentation

6.1 LCL_BooleanMatrix< N, M > Class Template Reference

`#include <LCL_BooleanMatrix.h>`

Public Member Functions

- LCL_BooleanMatrix ()
- LCL_BooleanMatrix (int in_r, int in_c)
- void print (ostream &in_OS=cout) const
- bool operator() (int i, int j) const
- bool & operator() (int i, int j)
- LCL_BooleanMatrix operator+ (const LCL_BooleanMatrix &in_Mat) const
- LCL_BooleanMatrix & operator+= (const LCL_BooleanMatrix &in_Mat)
- LCL_BooleanMatrix operator* (const LCL_BooleanMatrix &in_Mat) const
- LCL_BooleanMatrix & operator*= (const LCL_BooleanMatrix &in_Mat)

Public Attributes

- LCL_BoundedInt r
- LCL_BoundedInt c

6.1.1 Detailed Description

template<int N, int M>
class LCL_BooleanMatrix< N, M >

Definition at line 14 of file LCL_BooleanMatrix.h.

6.1.2 Constructor & Destructor Documentation

6.1.2.1 LCL_BooleanMatrix() [1/2]

```
template<int N, int M>
LCL_BooleanMatrix< N, M >::LCL_BooleanMatrix ( )
```

Definition at line 14 of file LCL_BooleanMatrix_imp1.h.

6.1.2.2 LCL_BooleanMatrix() [2/2]

```
template<int N, int M>
LCL_BooleanMatrix< N, M >::LCL_BooleanMatrix (
            int in_r,
            int in_c )
```

Definition at line 28 of file LCL_BooleanMatrix_imp1.h.

6.1.3 Member Function Documentation

6.1.3.1 operator()() [1/2]

```
template<int N, int M>
bool LCL_BooleanMatrix< N, M >::operator() (
            int i,
            int j ) const
```

Definition at line 41 of file LCL_BooleanMatrix_imp1.h.

6.1.3.2 operator()() [2/2]

```
template<int N, int M>
bool & LCL_BooleanMatrix< N, M >::operator() (
            int i,
            int j )
```

Definition at line 52 of file LCL_BooleanMatrix_imp1.h.

6.1.3.3 operator∗()

```
template<int N, int M>
LCL_BooleanMatrix< N, M > LCL_BooleanMatrix< N, M >::operator* (
            const LCL_BooleanMatrix< N, M > & in_Mat ) const
```

Definition at line 107 of file LCL_BooleanMatrix_imp1.h.

6.1.3.4 operator*=()

```
template<int N, int M>
LCL_BooleanMatrix& LCL_BooleanMatrix< N, M >::operator*= (
            const LCL_BooleanMatrix< N, M > & in_Mat )
```

6.1.3.5 operator+()

```
template<int N, int M>
LCL_BooleanMatrix< N, M > LCL_BooleanMatrix< N, M >::operator+ (
            const LCL_BooleanMatrix< N, M > & in_Mat ) const
```

Definition at line 72 of file LCL_BooleanMatrix_imp1.h.

6.1.3.6 operator+=()

```
template<int N, int M>
LCL_BooleanMatrix< N, M > & LCL_BooleanMatrix< N, M >::operator+= (
            const LCL_BooleanMatrix< N, M > & in_Mat )
```

Definition at line 92 of file LCL_BooleanMatrix_imp1.h.

6.1.3.7 print()

```
template<int N, int M>
void LCL_BooleanMatrix< N, M >::print (
            ostream & in_OS = cout ) const
```

Definition at line 62 of file LCL_BooleanMatrix_imp1.h.

6.1.4 Member Data Documentation

6.1.4.1 c

```
template<int N, int M>
LCL_BoundedInt LCL_BooleanMatrix< N, M >::c
```

Definition at line 17 of file LCL_BooleanMatrix.h.

6.1.4.2 r

```
template<int N, int M>
LCL_BoundedInt LCL_BooleanMatrix< N, M >::r
```

Definition at line 16 of file LCL_BooleanMatrix.h.

The documentation for this class was generated from the following files:

- LCL_BooleanMatrix.h
- LCL_BooleanMatrix_imp1.h

6.2 LCL_BoundedInt Class Reference

```
#include <LCL_BoundedInt.h>
```

Public Member Functions

- LCL_BoundedInt ()
- LCL_BoundedInt (int in_v, int in_min, int in_max)
- operator int () const
- LCL_BoundedInt & operator= (const int in_v)
- LCL_BoundedInt & operator() (const int in_min, const int in_max)

6.2.1 Detailed Description

Definition at line 4 of file LCL_BoundedInt.h.

6.2.2 Constructor & Destructor Documentation

6.2.2.1 LCL_BoundedInt() [1/2]

```
LCL_BoundedInt::LCL_BoundedInt ( )
```

Definition at line 5 of file LCL_BoundedInt.cpp.

6.2.2.2 LCL_BoundedInt() [2/2]

```
LCL_BoundedInt::LCL_BoundedInt (
            int in_v,
            int in_min,
            int in_max )
```

Definition at line 9 of file LCL_BoundedInt.cpp.

6.2.3 Member Function Documentation

6.2.3.1 operator int()

```
LCL_BoundedInt::operator int ( ) const  [inline]
```

Definition at line 14 of file LCL_BoundedInt.h.

6.2.3.2 operator()()

```
LCL_BoundedInt & LCL_BoundedInt::operator() (
            const int in_min,
            const int in_max )
```

Definition at line 23 of file LCL_BoundedInt.cpp.

6.2.3.3 operator=()

```
LCL_BoundedInt & LCL_BoundedInt::operator= (
            const int in_v )
```

Definition at line 15 of file LCL_BoundedInt.cpp.

The documentation for this class was generated from the following files:

- LCL_BoundedInt.h
- LCL_BoundedInt.cpp

6.3 LCL_Menu Class Reference

```
#include <LCL_Menu.h>
```

Public Member Functions

- LCL_Menu (int inMaxMenu, char inQuitKey, const char *inQuitMessage)
- ~LCL_Menu ()
- void run ()
- void setMenuTitle (const char *inMessage)
- void addMenuItem (char inKey, const char *inMessage, procPtr inFunc)

6.3.1 Detailed Description

Definition at line 10 of file LCL_Menu.h.

6.3.2 Constructor & Destructor Documentation

6.3.2.1 LCL_Menu()

```
LCL_Menu::LCL_Menu (
            int inMaxMenu,
            char inQuitKey,
            const char * inQuitMessage )
```

Definition at line 11 of file LCL_Menu.cpp.

6.3.2.2 ~LCL_Menu()

```
LCL_Menu::~LCL_Menu ( )
```

Definition at line 23 of file LCL_Menu.cpp.

6.3.3 Member Function Documentation

6.3.3.1 addMenuItem()

```
void LCL_Menu::addMenuItem (
            char inKey,
            const char * inMessage,
            procPtr inFunc )
```

Definition at line 65 of file LCL_Menu.cpp.

6.3.3.2 run()

```
void LCL_Menu::run ( )
```

Definition at line 34 of file LCL_Menu.cpp.

6.3.3.3 setMenuTitle()

```
void LCL_Menu::setMenuTitle (
            const char * inMessage )
```

Definition at line 61 of file LCL_Menu.cpp.

The documentation for this class was generated from the following files:

- LCL_Menu.h
- LCL_Menu.cpp

6.4 LCL_RealMatrix< N, M > Class Template Reference

```
#include <LCL_RealMatrix.h>
```

Public Member Functions

- LCL_RealMatrix ()
- void print (ostream &in_OS=cout) const
- double operator() (int i, int j) const
- double & operator() (int i, int j)

6.4.1 Detailed Description

template<int N, int M>
class LCL_RealMatrix< N, M >

Definition at line 15 of file LCL_RealMatrix.h.

6.4.2 Constructor & Destructor Documentation

6.4.2.1 LCL_RealMatrix()

```
template<int N, int M>
LCL_RealMatrix< N, M >::LCL_RealMatrix ( )
```

Definition at line 12 of file LCL_RealMatrix_imp1.h.

6.4.3 Member Function Documentation

6.4.3.1 operator()() [1/2]

```
template<int N, int M>
double LCL_RealMatrix< N, M >::operator() (
            int i,
            int j ) const
```

Definition at line 21 of file LCL_RealMatrix_imp1.h.

6.4.3.2 operator()() [2/2]

```
template<int N, int M>
double & LCL_RealMatrix< N, M >::operator() (
            int i,
            int j )
```

Definition at line 32 of file LCL_RealMatrix_imp1.h.

6.4.3.3 print()

```
template<int N, int M>
void LCL_RealMatrix< N, M >::print (
            ostream & in_OS = cout ) const
```

Definition at line 43 of file LCL_RealMatrix_imp1.h.

The documentation for this class was generated from the following files:

- LCL_RealMatrix.h
- LCL_RealMatrix_imp1.h

Chapter 7

File Documentation

7.1 LCL.h File Reference

```
#include "LCL_Bool.h"
#include "LCL_BooleanMatrix.h"
#include "LCL_BoundedInt.h"
#include "LCL_ConsoleIn.h"
#include "LCL_ConsoleOut.h"
#include "LCL_Int.h"
#include "LCL_Mat_GF2.h"
#include "LCL_Maths.h"
#include "LCL_RealMatrix.h"
```

Macros

- #define LCL_USE_LCL_NAMESPACES

7.1.1 Macro Definition Documentation

7.1.1.1 LCL_USE_LCL_NAMESPACES

```
#define LCL_USE_LCL_NAMESPACES
```

Definition at line 4 of file LCL.h.

7.2 LCL_Bool.cpp File Reference

```
#include "LCL_Bool.h"
#include <iostream>
#include <cmath>
#include <utility>
#include <cstring>
```

7.3 LCL_Bool.h File Reference

```
#include <iostream>
```

Namespaces

- LCL_Bool

Functions

- int LCL_Bool::IntToBoolVec (bool *out, int l, int m=-1)
- int LCL_Bool::BoolVecToInt (const bool *x, int len)
- int LCL_Bool::BitSize (int l)
- int LCL_Bool::Inner (bool *x1, bool *x2, int len)
- int LCL_Bool::Weight (bool *x, int len)
- void LCL_Bool::BitwiseAnd (bool *x1, bool *x2, bool *out, int len)
- void LCL_Bool::BitwiseXor (bool *x1, bool *x2, bool *out, int len)
- void LCL_Bool::copy (bool *src, bool *dst, int len)
- void LCL_Bool::print (bool *x, int len, const char *pre=NULL)
- bool LCL_Bool::increment (bool *x, int len)
- void LCL_Bool::zeros (bool *x, int len)
- bool LCL_Bool::nextUniquePerm (bool *out, const bool *in, int len)
- void LCL_Bool::fromString (bool *out, const char *in_s)
- int LCL_Bool::ReedDecoder (bool *x, int R, int M, bool *c=NULL, bool *e=NULL)

7.4 LCL_BooleanMatrix.h File Reference

```
#include <iostream>
#include <ostream>
#include "LCL_BoundedInt.h"
```

Classes

- class LCL_BooleanMatrix< N, M >

Variables

- const int LCL_SMALL = 1E1
- const int LCL_MEDIUM = 1E2
- const int LCL_LARGE = 1E3
- bool LCL_BOOLEAN_MATRIX_DUMMY

7.4.1 Variable Documentation

7.4.1.1 LCL_BOOLEAN_MATRIX_DUMMY

```
bool LCL_BOOLEAN_MATRIX_DUMMY
```

Definition at line 11 of file LCL_BooleanMatrix_imp1.h.

7.4.1.2 LCL_LARGE

```
const int LCL_LARGE = 1E3
```

Definition at line 52 of file LCL_BooleanMatrix.h.

7.4.1.3 LCL_MEDIUM

```
const int LCL_MEDIUM = 1E2
```

Definition at line 51 of file LCL_BooleanMatrix.h.

7.4.1.4 LCL_SMALL

```
const int LCL_SMALL = 1E1
```

Definition at line 50 of file LCL_BooleanMatrix.h.

7.5 LCL_BooleanMatrix_imp1.h File Reference

```
#include "LCL_BooleanMatrix.h"
#include <iostream>
#include <ostream>
#include "LCL_ConsoleOut.h"
```

Variables

- bool LCL_BOOLEAN_MATRIX_DUMMY = 0

7.5.1 Variable Documentation

7.5.1.1 LCL_BOOLEAN_MATRIX_DUMMY

```
bool LCL_BOOLEAN_MATRIX_DUMMY = 0
```

Definition at line 11 of file LCL_BooleanMatrix_imp1.h.

7.6 LCL_BoundedInt.cpp File Reference

```
#include <LCL_BoundedInt.h>
#include "LCL_ConsoleOut.h"
```

7.7 LCL_BoundedInt.h File Reference

Classes

- class LCL_BoundedInt

7.8 LCL_ConsoleIn.cpp File Reference

```
#include "LCL_ConsoleIn.h"
#include <iostream>
#include <stdio.h>
```

7.9 LCL_ConsoleIn.h File Reference

Namespaces

- LCL_ConsoleIn

Functions

- int LCL_ConsoleIn::GetCommandOutput (char *dest, int n, const char *comm)

  Retrieves the standard-out of a system command and puts it in a character array.

7.10 LCL_ConsoleOut.cpp File Reference

```
#include "LCL_ConsoleOut.h"
#include <iostream>
#include <ostream>
#include <ctime>
```

Variables

- int dout_n = 0

7.10.1 Variable Documentation

7.10.1.1 dout_n

```
int dout_n = 0
```

Definition at line 10 of file LCL_ConsoleOut.cpp.

7.11 LCL_ConsoleOut.h File Reference

```
#include <iostream>
#include <ostream>
#include <ctime>
```

Namespaces

- LCL_ConsoleOut

Macros

- #define FOut() if(g_output_file)(*g_output_file)

Functions

- ostream & LCL_ConsoleOut::LOut ()
- void LCL_ConsoleOut::dout ()
- void LCL_ConsoleOut::warning (const char *message, const char *function_name=NULL, const char *class_name=NULL)
- void LCL_ConsoleOut::error (const char *message, const char *function_name=NULL, const char *class↩_name=NULL)
- void LCL_ConsoleOut::comment (const char *message, const char *function_name=NULL, const char *class_name=NULL)
- double LCL_ConsoleOut::secs (clock_t tic, clock_t toc)

Variables

- int dout_n
- int LCL_ConsoleOut::LOut_Pad = 0

7.11.1 Macro Definition Documentation

7.11.1.1 FOut

```
#define FOut( ) if(g_output_file)(*g_output_file)
```

Definition at line 11 of file LCL_ConsoleOut.h.

7.11.2 Variable Documentation

7.11.2.1 dout_n

```
int dout_n
```

Definition at line 10 of file LCL_ConsoleOut.cpp.

7.12 LCL_Int.cpp File Reference

```
#include "LCL_Int.h"
#include <iostream>
#include <utility>
#include <cstdlib>
```

7.13 LCL_Int.h File Reference

```
#include <iostream>
```

Namespaces

- LCL_Int

  A collection of useful functions for integer arrays.

Functions

- void LCL_Int::sort (int *x, int n, bool desc=true, int *a=NULL, int method=0)

  Sorts int vector x of length n in descending (ascending) order.
- int LCL_Int::randi (int in_min, int in_max)
- void LCL_Int::randi (int *x, int n, int in_min, int in_max)
- void LCL_Int::print (int *x, int n, const char *pre=NULL)
- void LCL_Int::copy (int *dst, const int *src, int n)
- void LCL_Int::sub (int *dst, const int *src, int n, int m, int i0=0)

  Copies a sub-array of length m < n to a new array.
- void LCL_Int::concat (int *top, const int *bottom, int n, int m)

  Concatenates vector top of length n with bottom of length bottom, the result of which is stored in top.
- void LCL_Int::randperm (int *x, int n, int x0=0)

  Randomly permutes the input array x. Optionally adds a constant x0 to each element.

7.14 LCL_Mat_GF2.cpp File Reference

```
#include "LCL_Mat_GF2.h"
#include <iostream>
#include <ostream>
#include <cmath>
#include <cstdlib>
```

7.15 LCL_Mat_GF2.h File Reference

```
#include <iostream>
#include <ostream>
```

Namespaces

- LCL_Mat_GF2

Functions

- bool ** LCL_Mat_GF2::construct (int n, int m)
- void LCL_Mat_GF2::destruct (bool **A, int n, int m)
- void LCL_Mat_GF2::copy (bool **A, int n, int m, bool **O)
- void LCL_Mat_GF2::print (bool **A, int n, int m, char *pre=NULL, bool header=true, ostream &inOS=cout)
- void LCL_Mat_GF2::add (bool **A, bool **B, int n, int m, bool **O)
- void LCL_Mat_GF2::times (bool **A, bool **B, int n, int m, int p, bool **O)
- void LCL_Mat_GF2::transpose (bool **A, int n, int m, bool **O)
- void LCL_Mat_GF2::addrow (bool **A, int n, int m, int i_t, int i_s)
- void LCL_Mat_GF2::swaprow (bool **A, int n, int m, int i_1, int i_2)
- void LCL_Mat_GF2::addcol (bool **A, int n, int m, int j_t, int j_s)
- void LCL_Mat_GF2::swapcol (bool **A, int n, int m, int j_1, int j_2)
- void LCL_Mat_GF2::rowechelon (bool **A, int n, int m)
- bool ** LCL_Mat_GF2::nullspace (bool **A, int n, int m, int &d)
- void LCL_Mat_GF2::eye (bool **A, int n, int m)
- void LCL_Mat_GF2::zeros (bool **A, int n, int m)
- void LCL_Mat_GF2::random (bool **A, int n, int m)

7.16 LCL_Maths.cpp File Reference

```
#include "LCL_Maths.h"
#include <iostream>
#include <cmath>
```

7.17 LCL_Maths.h File Reference

Namespaces

- LCL_Maths

Functions

- unsigned long long int LCL_Maths::fact (int n)
- unsigned long long int LCL_Maths::nCr (int n, int r)

7.18 LCL_Menu.cpp File Reference

```
#include "LCL_Menu.h"
#include "LCL_MenuUtils.h"
#include <cstring>
#include <string>
#include <iostream>
```

7.19 LCL_Menu.h File Reference

```
#include <cstring>
#include <string>
#include <iostream>
```

Classes

- class LCL_Menu

7.20 LCL_MenuUtils.cpp File Reference

```
#include "LCL_MenuUtils.h"
#include <cstring>
#include <string>
#include <iostream>
```

7.21 LCL_MenuUtils.h File Reference

```
#include <string>
#include <iostream>
```

Namespaces

- LCL_MenuUtils

Functions

- void LCL_MenuUtils::getOneInt (int &inOut, istream &inIS, ostream &inOS, const char *inMessage)
- void LCL_MenuUtils::getOneDouble (double &inOut, istream &inIS, ostream &inOS, const char *inMessage)
- void LCL_MenuUtils::getOneChar (char &inOut, istream &inIS, ostream &inOS, const char *inMessage)
- void LCL_MenuUtils::getOneCString (char *inOut, istream &inIS, ostream &inOS, const char *inMessage)
- void LCL_MenuUtils::getOneString (string &inOut, istream &inIS, ostream &inOS, const char *inMessage)

7.22 LCL_RealMatrix.h File Reference

```
#include <iostream>
#include <ostream>
```

Classes

- class LCL_RealMatrix< N, M >

7.23 LCL_RealMatrix_imp1.h File Reference

```
#include "LCL_RealMatrix.h"
#include <iostream>
#include <ostream>
#include "LCL_ConsoleOut.h"
```

7.24 LCL_Utils.cpp File Reference

```
#include "LCL_Utils.h"
#include <cstdlib>
#include <cmath>
#include <iostream>
```

7.25 LCL_Utils.h File Reference

```
#include <cstdlib>
#include <cmath>
#include <iostream>
```

Namespaces

- LCL_Utils

  Contains handy functions.

Functions

- double LCL_Utils::rand_d ()

  Returns a random double between 0 and 1.
- int LCL_Utils::rand_i (int min, int max)

  Returns a random int between min and max.
- int LCL_Utils::rand_i (int num)

  Returns a random int between 0 and num.
- int LCL_Utils::flipBitN (int index, int Nbit)

  Flips the Nth bit of binary expansion of index.
- int LCL_Utils::kDelta (int i, int j)

  Returns 1 if and only if i==j.
- bool LCL_Utils::Bn (int index, int bit)

  Returns the Nth bit of binary expansion of index.
- void LCL_Utils::getComment (istream &inStr)

  Displays a commented out line from an input stream if it begins with one.
- int LCL_Utils::GCD (int inA, int inB)

  (I think) GCD calculates the Greatest Common Divisor between inA and inB.
- int LCL_Utils::factorize (int inC, int ∗inFactors)

  Factorizes an integer inC and places the factors in an array inFactors.

7.26 main.cpp File Reference

```
#include <iostream>
#include "LCL.h"
```

Functions

- int main (int argc, char ∗∗argv)

7.26.1 Function Documentation

7.26.1.1 main()

```
int main (
              int argc,
              char ** argv )
```

Definition at line 6 of file main.cpp.

7.27 mainpage.md File Reference

Index