Last update: July 7, 2020

Unit-tests

Unit Testing for the Modules of Mylib-C

Pierre L'Ecuyer, Robin Legault and Simon-Olivier Laperrière

Département d'Informatique et de Recherche Opérationnelle
Université de Montréal

This document describes a collection of unit tests for the modules of *Mylib-C*. It provides a set of functions to test the utility functions of the library. For each module defined in **mylib**, a corresponding module in **unit-tests** offers functions that test out the functions of said module. The module `unit` contains functions to test the whole library at the same time.

Copyright

Contents

unit

This module defines different macros and functions to produce the unit tests and ensure that the utility functions defined in the library are correct. It also contains functions allowing to execute all the unit tests of the library at the same time.

```
#include "gdef.h"
#include <stdio.h>
#include <string.h>
```

```
#define assert_double(n1, n2) util_Assert(compare(n1,n2), "Failure in double comparison"
```

This macro compares double values `n1` and `n2`. It returns an error message if these values do not correspond.

```
#define assert_int64_t(n1, n2) util_Assert(n1 == n2, "Failure in 64-bit comparison");
```

This macro compares 64-bit integer values `n1` and `n2`. It returns an error message if these values do not correspond.

```
#define assert_uint64_t(n1, n2) util_Assert(n1 == n2, "Failure in unsigned 64-bit
comparison");
```

This macro compares unsigned 64-bit integer values `n1` and `n2`. It returns an error message if these values do not correspond.

```
#define assert_int(n1, n2) util_Assert(n1 == n2, "Failure in int comparison");
```

This macro compares integer values `n1` and `n2`. It returns an error message if these values do not correspond.

```
#define assert_str(str1, str2) util_Assert(compare(strcmp(str1,str2),0), "Failure
in string comparison");
```

This macro compares strings `str1` and `str2`. It returns an error message if these values do not correspond.

```
lebool compare(double n1, double n2);
```

This function allow to compare double values `n1` and `n2` with acceptable size error less than 0.00001.

```
void unit_all();
```

This function runs all the unit tests of the library. It can be executed after the first compilation of the library to ensure that the installation process has worked without glitch.

num_utest

This module contains unit tests for the functions implemented in `num`.

`void num_utest_Round64();`

Unit testing of the function `num_Round64`.

`void num_utest_RoundD();`

Unit testing of the function `num_RoundD`.

`void num_utest_IsNumber();`

Unit testing of the function `num_IsNumber`.

`void num_utest_IntToStrBase();`

Unit testing of the function `num_IntToStrBase`.

`void num_utest_MultMod();`

Unit testing of the function `num_MultMod`.

`void num_utest_MultModDouble();`

Unit testing of the function `num_MultModDouble`.

`void num_utest_MultModDirect();`

Unit testing of the function `num_MultModDirect`.

`void num_utest_InvEuclid();`

Unit testing of the function `num_InvEuclid`.

`void num_utest_InvEuclid32();`

Unit testing of the function `num_InvEuclid32`.

`void num_utest_InvExpon();`

Unit testing of the function `num_InvExpon`.

`void num_utest_InvExpon32();`

Unit testing of the function `num_InvExpon32`.

`void num_utest_gcd();`

Unit testing of the function `num_gcd`.

`void num_utest_gcd32();`

Unit testing of the function `num_gcd32`.

`void num_utest_isMersennePrime();`

Unit testing of the function `num_isMersennePrime`.

`void num_utest_all();`

This function executes all the unit tests of `num`.

num2_utest

This module contains unit tests for the functions implemented in `num2`.

`void num2_utest_LnFactorial();`

Unit testing of the function `num2_LnFactorial`.

`void num2_utest_LnGamma();`

Unit testing of the function `num2_LnGamma`.

`void num2_utest_Combination();`

Unit testing of the function `num2_Combinaison`.

`void num2_utest_CalcMatStirling();`

Unit testing of the function `num2_CalcMatStirling`.

`void num2_utest_VolumeSphere();`

Unit testing of the function `num2_VolumeSphere`.

`void num2_utest_BesselK025();`

Unit testing of the function `num2_BesselK025`.

`void num2_utest_Digamma();`

Unit testing of the function `num2_Digamma`.

`void num2_utest_all();`

This function executes all the unit tests of `num2`.

bitset_utest

This module contains unit tests for the functions implemented in `bitset`.

`void bitset_utest_ReverseOrderSimple();`

Unit testing of the function `bitset_ReverseOrderSimple`.

`void bitset_utest_ReverseOrder();`

Unit testing of the function `bitset_ReverseOrder`.

`void bitset_utest_all();`

This function executes all the unit tests of `bitset`.

bitvector_utest

This module contains unit tests for the functions implemented in `bitvector`.

`void bitvector_utest_copy();`

Unit testing of the function `bitvector_copy`.

`void bitvector_utest_copyPart();`

Unit testing of the function `bitvector_copyPart`.

`void bitvector_utest_clearVector();`

Unit testing of the function `bitvector_clearVector`.

`void bitvector_utest_clearBit();`

Unit testing of the function `bitvector_clearBit`.

`void bitvector_utest_canonical();`

Unit testing of the function `bitvector_canonical`.

`void bitvector_utest_setAllOnes();`

Unit testing of the function `bitvector_setAllOnes`.

`void bitvector_utest_isZero();`

Unit testing of the function `bitvector_isZero`.

`void bitvector_utest_areEqual();`

Unit testing of the function `bitvector_areEqual`.

`void bitvector_utest_haveCommonBit();`

Unit testing of the function `bitvector_haveCommonBit`.

`void bitvector_utest_xor();`

Unit testing of the function `bitvector_xor`.

`void bitvector_utest_xor3();`

Unit testing of the function `bitvector_xor3`.

`void bitvector_utest_xorSelf();`

Unit testing of the function `bitvector_xorSelf`.

`void bitvector_utest_and();`

Unit testing of the function `bitvector_and`.

`void bitvector_utest_andSelf();`

Unit testing of the function `bitvector_andSelf`.

`void bitvector_utest_andMaskLow();`

Unit testing of the function `bitvector_andMaskLow`.

`void bitvector_utest_andInvMaskLow();`

Unit testing of the function `bitvector_andInvMaskLow`.

`void bitvector_utest_leftShift();`

Unit testing of the function `bitvector_leftShift`.

`void bitvector_utest_rightShift();`

Unit testing of the function `bitvector_rightShift`.

`void bitvector_utest_leftShiftSelf();`

Unit testing of the function `bitvector_leftShiftSelf`.

`void bitvector_utest_rightShiftSelf();`

Unit testing of the function `bitvector_rightShiftSelf`.

`void bitvector_utest_flip();`

Unit testing of the function `bitvector_flip`.

`void bitvector_utest_setMaskLow();`

Unit testing of the function `bitvector_setMaskLow`.

`void bitvector_utest_setInvMaskLow();`

Unit testing of the function `bitvector_setInvMaskLow`.

`void bitvector_utest_all();`

This function executes all the unit tests of `bitvector`.

bitmatrix_utest

This module contains unit tests for the functions implemented in `bitmatrix`.

`void bitmatrix_utest_copypart();`

Unit testing of the function `bitmatrix_copypart`.

`void bitmatrix_utest_copySpecial();`

Unit testing of the function `bitmatrix_copySpecial`.

`void bitmatrix_utest_transpose();`

Unit testing of the function `bitmatrix_transpose`.

`void bitmatrix_utest_exchangeRows();`

Unit testing of the function `bitmatrix_exchangeRows`.

`void bitmatrix_utest_xorVect();`

Unit testing of the function `bitmatrix_xorVect`.

`void bitmatrix_utest_diagonalize();`

Unit testing of the function `bitmatrix_diagonalize`.

`void bitmatrix_utest_gaussianElimination();`

Unit testing of the function `bitmatrix_gaussianElimination`.

`void bitmatrix_utest_specialGaussianElimination();`

Unit testing of the function `bitmatrix_specialGaussianElimination`.

`void bitmatrix_utest_completeElimination();`

Unit testing of the function `bitmatrix_completeElimination`.

`void bitmatrix_utest_inverse();`

Unit testing of the function `bitmatrix_inverse`.

`void bitmatrix_utest_productByVector();`

Unit testing of the function `bitmatrix_productByVector`.

`void bitmatrix_utest_product();`

Unit testing of the function `bitmatrix_product`.

`void bitmatrix_utest_power();`

Unit testing of the function `bitmatrix_power`.

`void bitmatrix_utest_powerOfTwo();`

Unit testing of the function `bitmatrix_powerOfTwo`.

`void bitmatrix_utest_all();`

This function executes all the unit tests of `bitmatrix`.

rngstream_utest

This module contains unit tests for the functions implemented in `rngstream`.

`void rngstream_utest_RandU01();`

Unit testing of the function `rngstream_RandU01`.

`void rngstream_utest_RandInt();`

Unit testing of the function `rngstream_RandInt`.

`void rngstream_utest_SetSeed();`

Unit testing of the function `rngstream_SetSeed`.

`void rngstream_utest_SetPackageSeed();`

Unit testing of the function `rngstream_SetPackageSeed`.

`void rngstream_utest_all();`

This function executes all the unit tests of `rngstream`.