hgmp

Claude Heiland-Allen

2015–2019

Contents

1 cbits/wrappers.c

```
#include <assert.h>
#include <HsFFI.h>
#include <ghc-gmp.h>

void mpz_set_HsInt(mpz_ptr dst, const HsInt n) {
  if (sizeof(HsInt) == sizeof(signed long int)) {
    mpz_set_si(dst, n);
  } else if (sizeof(HsInt) == sizeof(signed long int) + sizeof(unsigned long int↙
    ↳ )) {
    // Win64, see comments in integer-gmp/src/GHC/Integer/Type.hs
#define lobits (8 * sizeof(unsigned long int))
#define lomask (lobits - 1)
    mpz_set_si(dst, n >> lobits); // warns when branch will be unused
    mpz_mul_2exp(dst, dst, lobits);
    mpz_add_ui(dst, dst, n & lomask);
  } else {
    assert(! "supported HsInt size");
  }
}
```

2 CHANGELOG.md

```
# 0.1.1
- raw function foreign import bindings
- define hsc2hs #alignment macro for older GHC versions
```

```
# 0.1.0.1
- use hsc2hs's #alignment macro in Storable instances

# 0.1.0.0
- initial release
```

3 examples/primes.hs

```
{-# LANGUAGE ForeignFunctionInterface #-}
module Main (main) where

import Foreign.Ptr (Ptr(..))
import Numeric.GMP.Types (MPZ)
import Numeric.GMP.Utils (withInInteger, withOutInteger_)
import Numeric.GMP.Raw.Safe (mpz_nextprime)
import System.Environment (getArgs)
import System.IO.Unsafe (unsafePerformIO)

nextPrimeIO :: Integer -> IO Integer
nextPrimeIO n = do
  withOutInteger_ $ \rop ->
    withInInteger n $ \op ->
      mpz_nextprime rop op

nextPrime :: Integer -> Integer
nextPrime n = unsafePerformIO $ nextPrimeIO n

primes :: Integer -> [Integer]
primes = drop 1 . iterate nextPrime

main :: IO ()
main = do
  [sn] <- getArgs
  n <- readIO sn
  mapM_ print . take 10 . primes $ n
```

4 .gitignore

```
.cabal-sandbox/
cabal.sandbox.config
dist/
dist-newstyle/
```

5 hgmp.cabal

```
name:                hgmp
version:             0.1.1
synopsis:            Haskell interface to GMP
description:         Types and instances, and marshalling between Integer and
                     Rational and the corresponding GMP types, along with raw
                     foreign imports of GMP functions.  Allows FFI to GMP code
                     (whether in GMP itself or in third-party code that uses
                     GMP).
                     .
                     Supports only GHC with integer-gmp, this might change if
                     there's any demand.
```

```
homepage:            https://code.mathr.co.uk/hgmp

license:             BSD3
license-file:        LICENSE

author:              Claude Heiland-Allen
maintainer:          claude@mathr.co.uk
copyright:           2016,2017 Claude Heiland-Allen
category:            Numeric
build-type:          Simple
extra-source-files:  README.md CHANGELOG.md examples/primes.hs
cabal-version:       >=1.10

library
  exposed-modules:     Numeric.GMP.Utils
                     , Numeric.GMP.Types
                     , Numeric.GMP.Raw.Safe
                     , Numeric.GMP.Raw.Unsafe
  build-depends:       base >= 4.8 && < 4.14
                     , integer-gmp >= 1.0 && < 1.1
                     , ghc-prim >= 0.4 && < 0.6
  build-tools:         hsc2hs
  hs-source-dirs:      src
  c-sources:           cbits/wrappers.c
  default-language:    Haskell2010
  other-extensions:    DeriveDataTypeable
                       GeneralizedNewtypeDeriving
                       ForeignFunctionInterface
                       MagicHash
                       UnboxedTuples
  ghc-options:         -Wall

test-suite Main
  type:                exitcode-stdio-1.0
  hs-source-dirs:      tests
  main-is:             Main.hs
  build-depends:       base
                     , hgmp
                     , QuickCheck >= 2.8 && < 2.14
  default-language:    Haskell2010
  other-extensions:    ForeignFunctionInterface
                       TemplateHaskell

source-repository head
  type:                git
  location:            https://code.mathr.co.uk/hgmp.git

source-repository this
  type:                git
  location:            https://code.mathr.co.uk/hgmp.git
  tag:                 v0.1.1
```

6 LICENSE

```
Copyright (c)2016, Claude Heiland-Allen
```

7 README.md

hgmp

Haskell interface to GMP. Types and instances, and marshalling between Integer and Rational and the corresponding GMP types, along with raw foreign imports of GMP functions. Allows FFI to GMP code (whether in GMP itself or in third-party code that uses GMP).

A simple example illustrating binding to GMP's next probable-prime function:

```
{-# LANGUAGE ForeignFunctionInterface #-}

import Foreign.Ptr (Ptr(..))
import Numeric.GMP.Types (MPZ)
import Numeric.GMP.Utils (withInInteger, withOutInteger_)
import Numeric.GMP.Raw.Safe (mpz_nextprime)
import System.IO.Unsafe (unsafePerformIO)

nextPrime :: Integer -> Integer
nextPrime n =
  unsafePerformIO $
    withOutInteger_ $ \rop ->
      withInInteger n $ \op ->
        mpz_nextprime rop op
```

8 Setup.hs

```
import Distribution.Simple
main = defaultMain
```

9 src/Numeric/GMP/Raw/Safe.hs

```
{-# LANGUAGE ForeignFunctionInterface #-}
-- | Raw GMP foreign bindings, imported safe.
module Numeric.GMP.Raw.Safe where

import Foreign.Ptr (Ptr)
import Foreign.C.Types
import Numeric.GMP.Types

-- * Types for Documentation

type SrcPtr = Ptr
type VolatilePtr = Ptr
type VolatileSrcPtr = Ptr

-- * Integer Functions
-- ** Initialization Functions
foreign import ccall safe "__gmpz_init" mpz_init :: Ptr MPZ -> IO ()
foreign import ccall safe "__gmpz_init2" mpz_init2 :: Ptr MPZ -> MPBitCnt -> IO ()
foreign import ccall safe "__gmpz_clear" mpz_clear :: Ptr MPZ -> IO ()
foreign import ccall safe "__gmpz_realloc2" mpz_realloc2 :: Ptr MPZ -> MPBitCnt -> IO ()
-- ** Assignment Functions
foreign import ccall safe "__gmpz_set" mpz_set :: Ptr MPZ -> SrcPtr MPZ -> IO ()
foreign import ccall safe "__gmpz_set_ui" mpz_set_ui :: Ptr MPZ -> CULong -> IO ()
foreign import ccall safe "__gmpz_set_si" mpz_set_si :: Ptr MPZ -> CLong -> IO ()
foreign import ccall safe "__gmpz_set_d" mpz_set_d :: Ptr MPZ -> CDouble -> IO ()
foreign import ccall safe "__gmpz_set_q" mpz_set_q :: Ptr MPZ -> SrcPtr MPQ -> IO ()
foreign import ccall safe "__gmpz_set_f" mpz_set_f :: Ptr MPZ -> SrcPtr MPF -> IO ()
foreign import ccall safe "__gmpz_set_str" mpz_set_str :: Ptr MPZ -> SrcPtr CChar -> CInt -> IO CInt
foreign import ccall safe "__gmpz_swap" mpz_swap :: Ptr MPZ -> Ptr MPZ -> IO ()
-- ** Combined Initialization and Assignment Functions
foreign import ccall safe "__gmpz_init_set" mpz_init_set :: Ptr MPZ -> SrcPtr MPZ -> IO ()
foreign import ccall safe "__gmpz_init_set_ui" mpz_init_set_ui :: Ptr MPZ -> CULong -> IO ()
foreign import ccall safe "__gmpz_init_set_si" mpz_init_set_si :: Ptr MPZ -> CLong -> IO ()
foreign import ccall safe "__gmpz_init_set_d" mpz_init_set_d :: Ptr MPZ -> CDouble -> IO ()
foreign import ccall safe "__gmpz_init_set_str" mpz_init_set_str :: Ptr MPZ -> SrcPtr CChar -> CInt -> IO CInt
-- ** Conversion Functions
foreign import ccall safe "__gmpz_get_ui" mpz_get_ui :: SrcPtr MPZ -> IO CLong
```

```
foreign import ccall safe "__gmpz_get_si" mpz_get_si :: SrcPtr MPZ -> IO CLong
foreign import ccall safe "__gmpz_get_d" mpz_get_d :: SrcPtr MPZ -> IO CDouble
foreign import ccall safe "__gmpz_get_d_2exp" mpz_get_d_2exp :: Ptr CLong -> ↙
    ↳ SrcPtr MPZ -> IO CDouble
foreign import ccall safe "__gmpz_get_str" mpz_get_str :: Ptr CChar -> CInt -> ↙
    ↳ SrcPtr MPZ -> IO CChar
-- ** Arithmetic Functions
foreign import ccall safe "__gmpz_add" mpz_add :: Ptr MPZ -> SrcPtr MPZ -> ↙
    ↳ SrcPtr MPZ -> IO ()
foreign import ccall safe "__gmpz_add_ui" mpz_add_ui :: Ptr MPZ -> SrcPtr MPZ ->↙
    ↳  CULong -> IO ()
foreign import ccall safe "__gmpz_sub" mpz_sub :: Ptr MPZ -> SrcPtr MPZ -> ↙
    ↳ SrcPtr MPZ -> IO ()
foreign import ccall safe "__gmpz_sub_ui" mpz_sub_ui :: Ptr MPZ -> SrcPtr MPZ ->↙
    ↳  CULong -> IO ()
foreign import ccall safe "__gmpz_ui_sub" mpz_ui_sub :: Ptr MPZ -> CULong -> ↙
    ↳ SrcPtr MPZ -> IO ()
foreign import ccall safe "__gmpz_mul" mpz_mul :: Ptr MPZ -> SrcPtr MPZ -> ↙
    ↳ SrcPtr MPZ -> IO ()
foreign import ccall safe "__gmpz_mul_si" mpz_mul_si :: Ptr MPZ -> SrcPtr MPZ ->↙
    ↳  CLong -> IO ()
foreign import ccall safe "__gmpz_mul_ui" mpz_mul_ui :: Ptr MPZ -> SrcPtr MPZ ->↙
    ↳  CULong -> IO ()
foreign import ccall safe "__gmpz_addmul" mpz_addmul :: Ptr MPZ -> SrcPtr MPZ ->↙
    ↳  SrcPtr MPZ -> IO ()
foreign import ccall safe "__gmpz_addmul_ui" mpz_addmul_ui :: Ptr MPZ -> SrcPtr ↙
    ↳ MPZ -> CULong -> IO ()
foreign import ccall safe "__gmpz_submul" mpz_submul :: Ptr MPZ -> SrcPtr MPZ ->↙
    ↳  SrcPtr MPZ -> IO ()
foreign import ccall safe "__gmpz_submul_ui" mpz_submul_ui :: Ptr MPZ -> SrcPtr ↙
    ↳ MPZ -> CULong -> IO ()
foreign import ccall safe "__gmpz_mul_2exp" mpz_mul_2exp :: Ptr MPZ -> SrcPtr ↙
    ↳ MPZ -> MPBitCnt -> IO ()
-- mpz_neg
-- mpz_abs
-- ** Division Functions
foreign import ccall safe "__gmpz_cdiv_q" mpz_cdiv_q :: Ptr MPZ -> SrcPtr MPZ ->↙
    ↳  SrcPtr MPZ -> IO ()
foreign import ccall safe "__gmpz_cdiv_r" mpz_cdiv_r :: Ptr MPZ -> SrcPtr MPZ ->↙
    ↳  SrcPtr MPZ -> IO ()
foreign import ccall safe "__gmpz_cdiv_qr" mpz_cdiv_qr :: Ptr MPZ -> Ptr MPZ -> ↙
    ↳ SrcPtr MPZ -> SrcPtr MPZ -> IO ()
foreign import ccall safe "__gmpz_cdiv_q_ui" mpz_cdiv_q_ui :: Ptr MPZ -> SrcPtr ↙
    ↳ MPZ -> CULong -> IO CULong
foreign import ccall safe "__gmpz_cdiv_r_ui" mpz_cdiv_r_ui :: Ptr MPZ -> SrcPtr ↙
    ↳ MPZ -> CULong -> IO CULong
foreign import ccall safe "__gmpz_cdiv_qr_ui" mpz_cdiv_qr_ui :: Ptr MPZ -> Ptr ↙
    ↳ MPZ -> SrcPtr MPZ -> CULong -> IO CULong
foreign import ccall safe "__gmpz_cdiv_ui" mpz_cdiv_ui :: SrcPtr MPZ -> CULong ↙
    ↳ -> IO CULong
foreign import ccall safe "__gmpz_cdiv_q_2exp" mpz_cdiv_q_2exp :: Ptr MPZ -> ↙
    ↳ SrcPtr MPZ -> MPBitCnt -> IO ()
foreign import ccall safe "__gmpz_cdiv_r_2exp" mpz_cdiv_r_2exp :: Ptr MPZ -> ↙
    ↳ SrcPtr MPZ -> MPBitCnt -> IO ()
foreign import ccall safe "__gmpz_fdiv_q" mpz_fdiv_q :: Ptr MPZ -> SrcPtr MPZ ->↙
    ↳  SrcPtr MPZ -> IO ()
foreign import ccall safe "__gmpz_fdiv_r" mpz_fdiv_r :: Ptr MPZ -> SrcPtr MPZ ->↙
```

```
    ↪  SrcPtr MPZ -> IO ()
foreign import ccall safe "__gmpz_fdiv_qr" mpz_fdiv_qr :: Ptr MPZ -> Ptr MPZ -> ↙
    ↪ SrcPtr MPZ -> SrcPtr MPZ -> IO ()
foreign import ccall safe "__gmpz_fdiv_q_ui" mpz_fdiv_q_ui :: Ptr MPZ -> SrcPtr ↙
    ↪ MPZ -> CULong -> IO CULong
foreign import ccall safe "__gmpz_fdiv_r_ui" mpz_fdiv_r_ui :: Ptr MPZ -> SrcPtr ↙
    ↪ MPZ -> CULong -> IO CULong
foreign import ccall safe "__gmpz_fdiv_qr_ui" mpz_fdiv_qr_ui :: Ptr MPZ -> Ptr ↙
    ↪ MPZ -> SrcPtr MPZ -> CULong -> IO CULong
foreign import ccall safe "__gmpz_fdiv_ui" mpz_fdiv_ui :: SrcPtr MPZ -> CULong ↙
    ↪ -> IO CULong
foreign import ccall safe "__gmpz_fdiv_r_2exp" mpz_fdiv_r_2exp :: Ptr MPZ -> ↙
    ↪ SrcPtr MPZ -> MPBitCnt -> IO ()
foreign import ccall safe "__gmpz_fdiv_q_2exp" mpz_fdiv_q_2exp :: Ptr MPZ -> ↙
    ↪ SrcPtr MPZ -> MPBitCnt -> IO ()
foreign import ccall safe "__gmpz_tdiv_q" mpz_tdiv_q :: Ptr MPZ -> SrcPtr MPZ ->↙
    ↪  SrcPtr MPZ -> IO ()
foreign import ccall safe "__gmpz_tdiv_r" mpz_tdiv_r :: Ptr MPZ -> SrcPtr MPZ ->↙
    ↪  SrcPtr MPZ -> IO ()
foreign import ccall safe "__gmpz_tdiv_qr" mpz_tdiv_qr :: Ptr MPZ -> Ptr MPZ -> ↙
    ↪ SrcPtr MPZ -> SrcPtr MPZ -> IO ()
foreign import ccall safe "__gmpz_tdiv_q_ui" mpz_tdiv_q_ui :: Ptr MPZ -> SrcPtr ↙
    ↪ MPZ -> CULong -> IO CULong
foreign import ccall safe "__gmpz_tdiv_r_ui" mpz_tdiv_r_ui :: Ptr MPZ -> SrcPtr ↙
    ↪ MPZ -> CULong -> IO CULong
foreign import ccall safe "__gmpz_tdiv_qr_ui" mpz_tdiv_qr_ui :: Ptr MPZ -> Ptr ↙
    ↪ MPZ -> SrcPtr MPZ -> CULong -> IO CULong
foreign import ccall safe "__gmpz_tdiv_ui" mpz_tdiv_ui :: SrcPtr MPZ -> CULong ↙
    ↪ -> IO CULong
foreign import ccall safe "__gmpz_tdiv_q_2exp" mpz_tdiv_q_2exp :: Ptr MPZ -> ↙
    ↪ SrcPtr MPZ -> MPBitCnt -> IO ()
foreign import ccall safe "__gmpz_tdiv_r_2exp" mpz_tdiv_r_2exp :: Ptr MPZ -> ↙
    ↪ SrcPtr MPZ -> MPBitCnt -> IO ()
foreign import ccall safe "__gmpz_mod" mpz_mod :: Ptr MPZ -> SrcPtr MPZ -> ↙
    ↪ SrcPtr MPZ -> IO ()
-- mpz_mod_ui
foreign import ccall safe "__gmpz_divexact" mpz_divexact :: Ptr MPZ -> SrcPtr ↙
    ↪ MPZ -> SrcPtr MPZ -> IO ()
foreign import ccall safe "__gmpz_divexact_ui" mpz_divexact_ui :: Ptr MPZ -> ↙
    ↪ SrcPtr MPZ -> CULong -> IO ()
foreign import ccall safe "__gmpz_divisible_p" mpz_divisible_p :: SrcPtr MPZ -> ↙
    ↪ SrcPtr MPZ -> IO CInt
foreign import ccall safe "__gmpz_divisible_ui_p" mpz_divisible_ui_p :: SrcPtr ↙
    ↪ MPZ -> CULong -> IO CInt
foreign import ccall safe "__gmpz_divisible_2exp_p" mpz_divisible_2exp_p :: ↙
    ↪ SrcPtr MPZ -> MPBitCnt -> IO CInt
foreign import ccall safe "__gmpz_congruent_p" mpz_congruent_p :: SrcPtr MPZ -> ↙
    ↪ SrcPtr MPZ -> SrcPtr MPZ -> IO CInt
foreign import ccall safe "__gmpz_congruent_ui_p" mpz_congruent_ui_p :: SrcPtr ↙
    ↪ MPZ -> CULong -> CULong -> IO CInt
foreign import ccall safe "__gmpz_congruent_2exp_p" mpz_congruent_2exp_p :: ↙
    ↪ SrcPtr MPZ -> SrcPtr MPZ -> MPBitCnt -> IO CInt
-- ** Exponentiation Functions
foreign import ccall safe "__gmpz_powm" mpz_powm :: Ptr MPZ -> SrcPtr MPZ -> ↙
    ↪ SrcPtr MPZ -> SrcPtr MPZ -> IO ()
foreign import ccall safe "__gmpz_powm_ui" mpz_powm_ui :: Ptr MPZ -> SrcPtr MPZ ↙
    ↪ -> CULong -> SrcPtr MPZ -> IO ()
```

```
foreign import ccall safe "__gmpz_powm_sec" mpz_powm_sec :: Ptr MPZ -> SrcPtr ↙
    ↳ MPZ -> SrcPtr MPZ -> SrcPtr MPZ -> IO ()
foreign import ccall safe "__gmpz_pow_ui" mpz_pow_ui :: Ptr MPZ -> SrcPtr MPZ ->↙
    ↳  CULong -> IO ()
foreign import ccall safe "__gmpz_ui_pow_ui" mpz_ui_pow_ui :: Ptr MPZ -> CULong ↙
    ↳ -> CULong -> IO ()
-- ** Root Extraction Functions
foreign import ccall safe "__gmpz_root" mpz_root :: Ptr MPZ -> SrcPtr MPZ -> ↙
    ↳ CULong -> IO CInt
foreign import ccall safe "__gmpz_rootrem" mpz_rootrem :: Ptr MPZ -> Ptr MPZ -> ↙
    ↳ SrcPtr MPZ -> CULong -> IO ()
foreign import ccall safe "__gmpz_sqrt" mpz_sqrt :: Ptr MPZ -> SrcPtr MPZ -> IO ↙
    ↳ ()
foreign import ccall safe "__gmpz_sqrtrem" mpz_sqrtrem :: Ptr MPZ -> Ptr MPZ -> ↙
    ↳ SrcPtr MPZ -> IO ()
foreign import ccall safe "__gmpz_perfect_power_p" mpz_perfect_power_p :: SrcPtr↙
    ↳  MPZ -> IO CInt
-- mpz_perfect_square_p
-- ** Number Theoretic Functions
foreign import ccall safe "__gmpz_probab_prime_p" mpz_probab_prime_p :: SrcPtr ↙
    ↳ MPZ -> CInt -> IO CInt
foreign import ccall safe "__gmpz_nextprime" mpz_nextprime :: Ptr MPZ -> SrcPtr ↙
    ↳ MPZ -> IO ()
foreign import ccall safe "__gmpz_gcd" mpz_gcd :: Ptr MPZ -> SrcPtr MPZ -> ↙
    ↳ SrcPtr MPZ -> IO ()
foreign import ccall safe "__gmpz_gcd_ui" mpz_gcd_ui :: Ptr MPZ -> SrcPtr MPZ ->↙
    ↳  CULong -> IO CULong
foreign import ccall safe "__gmpz_gcdext" mpz_gcdext :: Ptr MPZ -> Ptr MPZ -> ↙
    ↳ Ptr MPZ -> SrcPtr MPZ -> SrcPtr MPZ -> IO ()
foreign import ccall safe "__gmpz_lcm" mpz_lcm :: Ptr MPZ -> SrcPtr MPZ -> ↙
    ↳ SrcPtr MPZ -> IO ()
foreign import ccall safe "__gmpz_lcm_ui" mpz_lcm_ui :: Ptr MPZ -> SrcPtr MPZ ->↙
    ↳  CULong -> IO ()
foreign import ccall safe "__gmpz_invert" mpz_invert :: Ptr MPZ -> SrcPtr MPZ ->↙
    ↳  SrcPtr MPZ -> IO CInt
foreign import ccall safe "__gmpz_jacobi" mpz_jacobi :: SrcPtr MPZ -> SrcPtr MPZ↙
    ↳  -> IO CInt
-- mpz_legendre
-- mpz_kronecker
foreign import ccall safe "__gmpz_kronecker_si" mpz_kronecker_si :: SrcPtr MPZ ↙
    ↳ -> CLong -> IO CInt
foreign import ccall safe "__gmpz_kronecker_ui" mpz_kronecker_ui :: SrcPtr MPZ ↙
    ↳ -> CULong -> IO CInt
foreign import ccall safe "__gmpz_si_kronecker" mpz_si_kronecker :: CLong -> ↙
    ↳ SrcPtr MPZ -> IO CInt
foreign import ccall safe "__gmpz_ui_kronecker" mpz_ui_kronecker :: CULong -> ↙
    ↳ SrcPtr MPZ -> IO CInt
foreign import ccall safe "__gmpz_remove" mpz_remove :: Ptr MPZ -> SrcPtr MPZ ->↙
    ↳  SrcPtr MPZ -> IO MPBitCnt
foreign import ccall safe "__gmpz_fac_ui" mpz_fac_ui :: Ptr MPZ -> CULong -> IO ↙
    ↳ ()
foreign import ccall safe "__gmpz_2fac_ui" mpz_2fac_ui :: Ptr MPZ -> CULong -> ↙
    ↳ IO ()
foreign import ccall safe "__gmpz_mfac_uiui" mpz_mfac_uiui :: Ptr MPZ -> CULong ↙
    ↳ -> CULong -> IO ()
foreign import ccall safe "__gmpz_primorial_ui" mpz_primorial_ui :: Ptr MPZ -> ↙
    ↳ CULong -> IO ()
```

```
130 foreign import ccall safe "__gmpz_bin_ui" mpz_bin_ui :: Ptr MPZ -> SrcPtr MPZ ->↙
      ↳ CULong -> IO ()
    foreign import ccall safe "__gmpz_bin_uiui" mpz_bin_uiui :: Ptr MPZ -> CULong ->↙
      ↳ CULong -> IO ()
    foreign import ccall safe "__gmpz_fib_ui" mpz_fib_ui :: Ptr MPZ -> CULong -> IO ↙
      ↳ ()
    foreign import ccall safe "__gmpz_fib2_ui" mpz_fib2_ui :: Ptr MPZ -> Ptr MPZ -> ↙
      ↳ CULong -> IO ()
    foreign import ccall safe "__gmpz_lucnum_ui" mpz_lucnum_ui :: Ptr MPZ -> CULong ↙
      ↳ -> IO ()
135 foreign import ccall safe "__gmpz_lucnum2_ui" mpz_lucnum2_ui :: Ptr MPZ -> Ptr ↙
      ↳ MPZ -> CULong -> IO ()
    -- ** Comparison Functions
    foreign import ccall safe "__gmpz_cmp" mpz_cmp :: SrcPtr MPZ -> SrcPtr MPZ -> IO↙
      ↳ CInt
    foreign import ccall safe "__gmpz_cmp_d" mpz_cmp_d :: SrcPtr MPZ -> CDouble -> ↙
      ↳ IO CInt
    foreign import ccall safe "__gmpz_cmp_si" mpz_cmp_si :: SrcPtr MPZ -> CLong -> ↙
      ↳ IO CInt
140 foreign import ccall safe "__gmpz_cmp_ui" mpz_cmp_ui :: SrcPtr MPZ -> CULong -> ↙
      ↳ IO CInt
    foreign import ccall safe "__gmpz_cmpabs" mpz_cmpabs :: SrcPtr MPZ -> SrcPtr MPZ↙
      ↳ -> IO CInt
    foreign import ccall safe "__gmpz_cmpabs_d" mpz_cmpabs_d :: SrcPtr MPZ -> ↙
      ↳ CDouble -> IO CInt
    foreign import ccall safe "__gmpz_cmpabs_ui" mpz_cmpabs_ui :: SrcPtr MPZ -> ↙
      ↳ CULong -> IO CInt
    -- mpz_sgn
145 -- ** Logical and Bit Manipulation Functions
    foreign import ccall safe "__gmpz_and" mpz_and :: Ptr MPZ -> SrcPtr MPZ -> ↙
      ↳ SrcPtr MPZ -> IO ()
    foreign import ccall safe "__gmpz_ior" mpz_ior :: Ptr MPZ -> SrcPtr MPZ -> ↙
      ↳ SrcPtr MPZ -> IO ()
    foreign import ccall safe "__gmpz_xor" mpz_xor :: Ptr MPZ -> SrcPtr MPZ -> ↙
      ↳ SrcPtr MPZ -> IO ()
    foreign import ccall safe "__gmpz_com" mpz_com :: Ptr MPZ -> SrcPtr MPZ -> IO ()
150 -- mpz_popcount
    foreign import ccall safe "__gmpz_hamdist" mpz_hamdist :: SrcPtr MPZ -> SrcPtr ↙
      ↳ MPZ -> IO MPBitCnt
    foreign import ccall safe "__gmpz_scan0" mpz_scan0 :: SrcPtr MPZ -> MPBitCnt -> ↙
      ↳ IO MPBitCnt
    foreign import ccall safe "__gmpz_scan1" mpz_scan1 :: SrcPtr MPZ -> MPBitCnt -> ↙
      ↳ IO MPBitCnt
    foreign import ccall safe "__gmpz_setbit" mpz_setbit :: Ptr MPZ -> MPBitCnt -> ↙
      ↳ IO ()
155 foreign import ccall safe "__gmpz_clrbit" mpz_clrbit :: Ptr MPZ -> MPBitCnt -> ↙
      ↳ IO ()
    foreign import ccall safe "__gmpz_combit" mpz_combit :: Ptr MPZ -> MPBitCnt -> ↙
      ↳ IO ()
    foreign import ccall safe "__gmpz_tstbit" mpz_tstbit :: SrcPtr MPZ -> MPBitCnt ↙
      ↳ -> IO CInt
    -- ** Input and Output Functions
    foreign import ccall safe "__gmpz_out_str" mpz_out_str :: Ptr CFile -> CInt -> ↙
      ↳ SrcPtr MPZ -> IO CSize
160 foreign import ccall safe "__gmpz_inp_str" mpz_inp_str :: Ptr MPZ -> Ptr CFile ↙
      ↳ -> CInt -> IO CSize
    foreign import ccall safe "__gmpz_out_raw" mpz_out_raw :: Ptr CFile -> SrcPtr ↙
```

```
    ↳ MPZ -> IO CSize
foreign import ccall safe "__gmpz_inp_raw" mpz_inp_raw :: Ptr MPZ -> Ptr CFile ↙
    ↳ -> IO CSize
-- ** Random Number Functions
foreign import ccall safe "__gmpz_urandomb" mpz_urandomb :: Ptr MPZ -> Ptr ↙
    ↳ GMPRandState -> MPBitCnt -> IO ()
foreign import ccall safe "__gmpz_urandomm" mpz_urandomm :: Ptr MPZ -> Ptr ↙
    ↳ GMPRandState -> SrcPtr MPZ -> IO ()
foreign import ccall safe "__gmpz_rrandomb" mpz_rrandomb :: Ptr MPZ -> Ptr ↙
    ↳ GMPRandState -> MPBitCnt -> IO ()
-- ** Integer Import and Export
foreign import ccall safe "__gmpz_import" mpz_import :: Ptr MPZ -> CSize -> CInt↙
    ↳  -> CSize -> CInt -> CSize -> Ptr a -> IO ()
foreign import ccall safe "__gmpz_export" mpz_export :: Ptr a -> Ptr CSize -> ↙
    ↳ CInt -> CSize -> CInt -> CSize -> SrcPtr MPZ -> IO ()
-- ** Miscellaneous Functions
-- mpz_fits_ulong_p
foreign import ccall safe "__gmpz_fits_slong_p" mpz_fits_slong_p :: SrcPtr MPZ ↙
    ↳ -> IO CInt
-- mpz_fits_uint_p
foreign import ccall safe "__gmpz_fits_sint_p" mpz_fits_sint_p :: SrcPtr MPZ -> ↙
    ↳ IO CInt
-- mpz_fits_ushort_p
foreign import ccall safe "__gmpz_fits_sshort_p" mpz_fits_sshort_p :: SrcPtr MPZ↙
    ↳  -> IO CInt
-- mpz_odd_p
-- mpz_even_p
foreign import ccall safe "__gmpz_sizeinbase" mpz_sizeinbase :: SrcPtr MPZ -> ↙
    ↳ CInt -> IO CSize
-- ** Special Functions
foreign import ccall safe "__gmpz_realloc" mpz_realloc :: Ptr MPZ -> MPSize -> ↙
    ↳ IO ()
-- mpz_get_limbn
-- mpz_size
foreign import ccall safe "__gmpz_limbs_read" mpz_limbs_read :: SrcPtr MPZ -> IO↙
    ↳  (SrcPtr MPLimb)
foreign import ccall safe "__gmpz_limbs_write" mpz_limbs_write :: Ptr MPZ -> ↙
    ↳ MPSize -> IO (Ptr MPLimb)
foreign import ccall safe "__gmpz_limbs_modify" mpz_limbs_modify :: Ptr MPZ -> ↙
    ↳ MPSize -> IO (Ptr MPLimb)
foreign import ccall safe "__gmpz_limbs_finish" mpz_limbs_finish :: Ptr MPZ -> ↙
    ↳ MPSize -> IO ()
foreign import ccall safe "__gmpz_roinit_n" mpz_roinit_n :: Ptr MPZ -> SrcPtr ↙
    ↳ MPLimb -> MPSize -> IO (SrcPtr MPZ)
-- MPZ_ROINIT_N

-- * Rational Number Functions
foreign import ccall safe "__gmpq_canonicalize" mpq_canonicalize :: Ptr MPQ -> ↙
    ↳ IO ()
-- ** Initialization and Assignment Functions
foreign import ccall safe "__gmpq_init" mpq_init :: Ptr MPQ -> IO ()
foreign import ccall safe "__gmpq_clear" mpq_clear :: Ptr MPQ -> IO ()
foreign import ccall safe "__gmpq_set" mpq_set :: Ptr MPQ -> SrcPtr MPQ -> IO ()
foreign import ccall safe "__gmpq_set_z" mpq_set_z :: Ptr MPQ -> SrcPtr MPZ -> ↙
    ↳ IO ()
foreign import ccall safe "__gmpq_set_ui" mpq_set_ui :: Ptr MPQ -> CULong -> ↙
    ↳ CULong -> IO ()
```

```
foreign import ccall safe "__gmpq_set_si" mpq_set_si :: Ptr MPQ -> CLong -> ↙
    ↳ CULong -> IO ()
foreign import ccall safe "__gmpq_set_str" mpq_set_str :: Ptr MPQ -> SrcPtr ↙
    ↳ CChar -> CInt -> IO CInt
foreign import ccall safe "__gmpq_swap" mpq_swap :: Ptr MPQ -> Ptr MPQ -> IO ()
-- ** Conversion Functions
foreign import ccall safe "__gmpq_get_d" mpq_get_d :: SrcPtr MPQ -> IO CDouble
foreign import ccall safe "__gmpq_set_d" mpq_set_d :: Ptr MPQ -> CDouble -> IO ↙
    ↳ ()
foreign import ccall safe "__gmpq_set_f" mpq_set_f :: Ptr MPQ -> SrcPtr MPF -> ↙
    ↳ IO ()
foreign import ccall safe "__gmpq_get_str" mpq_get_str :: Ptr CChar -> CInt -> ↙
    ↳ SrcPtr MPQ -> IO (Ptr CChar)
-- ** Arithmetic Functions
foreign import ccall safe "__gmpq_add" mpq_add :: Ptr MPQ -> SrcPtr MPQ -> ↙
    ↳ SrcPtr MPQ -> IO ()
foreign import ccall safe "__gmpq_sub" mpq_sub :: Ptr MPQ -> SrcPtr MPQ -> ↙
    ↳ SrcPtr MPQ -> IO ()
foreign import ccall safe "__gmpq_mul" mpq_mul :: Ptr MPQ -> SrcPtr MPQ -> ↙
    ↳ SrcPtr MPQ -> IO ()
foreign import ccall safe "__gmpq_mul_2exp" mpq_mul_2exp :: Ptr MPQ -> SrcPtr ↙
    ↳ MPQ -> MPBitCnt -> IO ()
foreign import ccall safe "__gmpq_div" mpq_div :: Ptr MPQ -> SrcPtr MPQ -> ↙
    ↳ SrcPtr MPQ -> IO ()
foreign import ccall safe "__gmpq_div_2exp" mpq_div_2exp :: Ptr MPQ -> SrcPtr ↙
    ↳ MPQ -> MPBitCnt -> IO ()
-- mpq_neg
-- mpq_abs
foreign import ccall safe "__gmpq_inv" mpq_inv :: Ptr MPQ -> SrcPtr MPQ -> IO ()
-- ** Comparison Functions
foreign import ccall safe "__gmpq_cmp" mpq_cmp :: SrcPtr MPQ -> SrcPtr MPQ -> IO↙
    ↳  CInt
foreign import ccall safe "__gmpq_cmp_z" mpq_cmp_z :: SrcPtr MPQ -> SrcPtr MPZ ↙
    ↳ -> IO CInt
foreign import ccall safe "__gmpq_cmp_ui" mpq_cmp_ui :: SrcPtr MPQ -> CULong -> ↙
    ↳ CULong -> IO CInt
foreign import ccall safe "__gmpq_cmp_si" mpq_cmp_si :: SrcPtr MPQ -> CLong -> ↙
    ↳ CULong -> IO CInt
-- mpq_sgn
foreign import ccall safe "__gmpq_equal" mpq_equal :: SrcPtr MPQ -> SrcPtr MPQ ↙
    ↳ -> IO CInt
-- ** Applying Integer Functions to Rationals
-- See also 'mpq_numref' and 'mpq_denref'.
foreign import ccall safe "__gmpq_get_num" mpq_get_num :: Ptr MPZ -> SrcPtr MPQ ↙
    ↳ -> IO ()
foreign import ccall safe "__gmpq_get_den" mpq_get_den :: Ptr MPZ -> SrcPtr MPQ ↙
    ↳ -> IO ()
foreign import ccall safe "__gmpq_set_num" mpq_set_num :: Ptr MPQ -> SrcPtr MPZ ↙
    ↳ -> IO ()
foreign import ccall safe "__gmpq_set_den" mpq_set_den :: Ptr MPQ -> SrcPtr MPZ ↙
    ↳ -> IO ()
-- ** Input and Output Functions
foreign import ccall safe "__gmpq_out_str" mpq_out_str :: Ptr CFile -> CInt -> ↙
    ↳ SrcPtr MPQ -> IO CSize
foreign import ccall safe "__gmpq_inp_str" mpq_inp_str :: Ptr MPQ -> Ptr CFile ↙
    ↳ -> CInt -> IO CSize
```

```
-- * Floating-point Functions
-- ** Initialization Functions
{-
-- not thread-safe, ie, requires running everything in a bound thread for ↙
    ↳ expected behaviour
foreign import ccall safe "__gmpf_set_default_prec" mpf_set_default_prec :: ↙
    ↳ MPBitCnt -> IO ()
foreign import ccall safe "__gmpf_get_default_prec" mpf_get_default_prec :: IO ↙
    ↳ MPBitCnt
foreign import ccall safe "__gmpf_init" mpf_init :: Ptr MPF -> IO ()
-}
foreign import ccall safe "__gmpf_init2" mpf_init2 :: Ptr MPF -> MPBitCnt -> IO ↙
    ↳ ()
foreign import ccall safe "__gmpf_clear" mpf_clear :: Ptr MPF -> IO ()
foreign import ccall safe "__gmpf_get_prec" mpf_get_prec :: SrcPtr MPF -> IO ↙
    ↳ MPBitCnt
foreign import ccall safe "__gmpf_set_prec" mpf_set_prec :: Ptr MPF -> MPBitCnt ↙
    ↳ -> IO ()
foreign import ccall safe "__gmpf_set_prec_raw" mpf_set_prec_raw :: Ptr MPF -> ↙
    ↳ MPBitCnt -> IO ()
-- ** Assignment Functions
foreign import ccall safe "__gmpf_set" mpf_set :: Ptr MPF -> SrcPtr MPF -> IO ()
foreign import ccall safe "__gmpf_set_ui" mpf_set_ui :: Ptr MPF -> CULong -> IO ↙
    ↳ ()
foreign import ccall safe "__gmpf_set_si" mpf_set_si :: Ptr MPF -> CLong -> IO ↙
    ↳ ()
foreign import ccall safe "__gmpf_set_d" mpf_set_d :: Ptr MPF -> CDouble -> IO ↙
    ↳ ()
foreign import ccall safe "__gmpf_set_z" mpf_set_z :: Ptr MPF -> SrcPtr MPZ -> ↙
    ↳ IO ()
foreign import ccall safe "__gmpf_set_q" mpf_set_q :: Ptr MPF -> SrcPtr MPQ -> ↙
    ↳ IO ()
foreign import ccall safe "__gmpf_set_str" mpf_set_str :: Ptr MPF -> SrcPtr ↙
    ↳ CChar -> CInt -> IO CInt
foreign import ccall safe "__gmpf_swap" mpf_swap :: Ptr MPF -> Ptr MPF -> IO ()
-- ** Combined Initialization and Assignment Functions
foreign import ccall safe "__gmpf_init_set" mpf_init_set :: Ptr MPF -> SrcPtr ↙
    ↳ MPF -> IO ()
foreign import ccall safe "__gmpf_init_set_ui" mpf_init_set_ui :: Ptr MPF -> ↙
    ↳ CULong -> IO ()
foreign import ccall safe "__gmpf_init_set_si" mpf_init_set_si :: Ptr MPF -> ↙
    ↳ CLong -> IO ()
foreign import ccall safe "__gmpf_init_set_d" mpf_init_set_d :: Ptr MPF -> ↙
    ↳ CDouble -> IO ()
foreign import ccall safe "__gmpf_init_set_str" mpf_init_set_str :: Ptr MPF -> ↙
    ↳ SrcPtr CChar -> CInt -> IO CInt
-- ** Conversion Functions
foreign import ccall safe "__gmpf_get_d" mpf_get_d :: SrcPtr MPF -> IO CDouble
foreign import ccall safe "__gmpf_get_d_2exp" mpf_get_d_2exp :: Ptr CLong -> ↙
    ↳ SrcPtr MPF -> IO CDouble
foreign import ccall safe "__gmpf_get_si" mpf_get_si :: SrcPtr MPF -> IO CLong
foreign import ccall safe "__gmpf_get_ui" mpf_get_ui :: SrcPtr MPF -> IO CULong
foreign import ccall safe "__gmpf_get_str" mpf_get_str :: Ptr CChar -> Ptr MPExp↙
    ↳  -> CInt -> CSize -> SrcPtr MPF -> IO (Ptr CChar)
-- ** Arithmetic Functions
foreign import ccall safe "__gmpf_add" mpf_add :: Ptr MPF -> SrcPtr MPF -> ↙
    ↳ SrcPtr MPF -> IO ()
```

```
270 foreign import ccall safe "__gmpf_add_ui" mpf_add_ui :: Ptr MPF -> SrcPtr MPF ->↙
      ↳ CULong -> IO ()
    foreign import ccall safe "__gmpf_sub" mpf_sub :: Ptr MPF -> SrcPtr MPF -> ↙
      ↳ SrcPtr MPF -> IO ()
    foreign import ccall safe "__gmpf_sub_ui" mpf_sub_ui :: Ptr MPF -> SrcPtr MPF ->↙
      ↳ CULong -> IO ()
    foreign import ccall safe "__gmpf_ui_sub" mpf_ui_sub :: Ptr MPF -> CULong -> ↙
      ↳ SrcPtr MPF -> IO ()
    foreign import ccall safe "__gmpf_mul" mpf_mul :: Ptr MPF -> SrcPtr MPF -> ↙
      ↳ SrcPtr MPF -> IO ()
275 foreign import ccall safe "__gmpf_mul_ui" mpf_mul_ui :: Ptr MPF -> SrcPtr MPF ->↙
      ↳ CULong -> IO ()
    foreign import ccall safe "__gmpf_div" mpf_div :: Ptr MPF -> SrcPtr MPF -> ↙
      ↳ SrcPtr MPF -> IO ()
    foreign import ccall safe "__gmpf_ui_div" mpf_ui_div :: Ptr MPF -> CULong -> ↙
      ↳ SrcPtr MPF -> IO ()
    foreign import ccall safe "__gmpf_div_ui" mpf_div_ui :: Ptr MPF -> SrcPtr MPF ->↙
      ↳ CULong -> IO ()
    foreign import ccall safe "__gmpf_sqrt" mpf_sqrt :: Ptr MPF -> SrcPtr MPF -> IO ↙
      ↳ ()
280 foreign import ccall safe "__gmpf_sqrt_ui" mpf_sqrt_ui :: Ptr MPF -> CULong -> ↙
      ↳ IO ()
    foreign import ccall safe "__gmpf_pow_ui" mpf_pow_ui :: Ptr MPF -> SrcPtr MPF ->↙
      ↳ CULong -> IO ()
    foreign import ccall safe "__gmpf_neg" mpf_neg :: Ptr MPF -> SrcPtr MPF -> IO ()
    foreign import ccall safe "__gmpf_abs" mpf_abs :: Ptr MPF -> SrcPtr MPF -> IO ()
    foreign import ccall safe "__gmpf_mul_2exp" mpf_mul_2exp :: Ptr MPF -> SrcPtr ↙
      ↳ MPF -> MPBitCnt -> IO ()
285 foreign import ccall safe "__gmpf_div_2exp" mpf_div_2exp :: Ptr MPF -> SrcPtr ↙
      ↳ MPF -> MPBitCnt -> IO ()
    -- ** Comparison Functions
    foreign import ccall safe "__gmpf_cmp" mpf_cmp :: SrcPtr MPF -> SrcPtr MPF -> IO↙
      ↳ CInt
    foreign import ccall safe "__gmpf_cmp_z" mpf_cmp_z :: SrcPtr MPF -> SrcPtr MPZ ↙
      ↳ -> IO CInt
    foreign import ccall safe "__gmpf_cmp_d" mpf_cmp_d :: SrcPtr MPF -> CDouble -> ↙
      ↳ IO CInt
290 foreign import ccall safe "__gmpf_cmp_ui" mpf_cmp_ui :: SrcPtr MPF -> CULong -> ↙
      ↳ IO CInt
    foreign import ccall safe "__gmpf_cmp_si" mpf_cmp_si :: SrcPtr MPF -> CLong -> ↙
      ↳ IO CInt
    foreign import ccall safe "__gmpf_reldiff" mpf_reldiff :: Ptr MPF -> SrcPtr MPF ↙
      ↳ -> SrcPtr MPF -> IO ()
    -- mpf_sgn
    -- ** Input and Output Functions
295 foreign import ccall safe "__gmpf_out_str" mpf_out_str :: Ptr CFile -> CInt -> ↙
      ↳ CSize -> SrcPtr MPF -> IO CSize
    foreign import ccall safe "__gmpf_inp_str" mpf_inp_str :: Ptr MPF -> Ptr CFile ↙
      ↳ -> CInt -> IO CSize
    -- ** Miscellaneous Functions
    foreign import ccall safe "__gmpf_ceil" mpf_ceil :: Ptr MPF -> SrcPtr MPF -> IO ↙
      ↳ ()
    foreign import ccall safe "__gmpf_floor" mpf_floor :: Ptr MPF -> SrcPtr MPF -> ↙
      ↳ IO ()
300 foreign import ccall safe "__gmpf_trunc" mpf_trunc :: Ptr MPF -> SrcPtr MPF -> ↙
      ↳ IO ()
    foreign import ccall safe "__gmpf_integer_p" mpf_integer_p :: SrcPtr MPF -> IO ↙
```

```
    ↳ CInt
foreign import ccall safe "__gmpf_fits_ulong_p" mpf_fits_ulong_p :: SrcPtr MPF ↙
    ↳ -> IO CInt
foreign import ccall safe "__gmpf_fits_slong_p" mpf_fits_slong_p :: SrcPtr MPF ↙
    ↳ -> IO CInt
foreign import ccall safe "__gmpf_fits_uint_p" mpf_fits_uint_p :: SrcPtr MPF -> ↙
    ↳ IO CInt
foreign import ccall safe "__gmpf_fits_sint_p" mpf_fits_sint_p :: SrcPtr MPF -> ↙
    ↳ IO CInt
foreign import ccall safe "__gmpf_fits_ushort_p" mpf_fits_ushort_p :: SrcPtr MPF↙
    ↳  -> IO CInt
foreign import ccall safe "__gmpf_fits_sshort_p" mpf_fits_sshort_p :: SrcPtr MPF↙
    ↳  -> IO CInt
foreign import ccall safe "__gmpf_urandomb" mpf_urandomb :: Ptr MPF -> Ptr ↙
    ↳ GMPRandState -> MPBitCnt -> IO ()
foreign import ccall safe "__gmpf_random2" mpf_random2 :: Ptr MPF -> MPSize -> ↙
    ↳ MPExp -> IO ()

-- * Random Number Functions
-- ** Random State Initialization
foreign import ccall safe "__gmp_randinit_default" gmp_randinit_default :: Ptr ↙
    ↳ GMPRandState -> IO ()
foreign import ccall safe "__gmp_randinit_mt" gmp_randinit_mt :: Ptr ↙
    ↳ GMPRandState -> IO ()
foreign import ccall safe "__gmp_randinit_lc_2exp" gmp_randinit_lc_2exp :: Ptr ↙
    ↳ GMPRandState -> SrcPtr MPZ -> CULong -> MPBitCnt -> IO ()
foreign import ccall safe "__gmp_randinit_lc_2exp_size" ↙
    ↳ gmp_randinit_lc_2exp_size :: Ptr GMPRandState -> MPBitCnt -> IO CInt
foreign import ccall safe "__gmp_randinit_set" gmp_randinit_set :: Ptr ↙
    ↳ GMPRandState -> SrcPtr GMPRandState -> IO ()
foreign import ccall safe "__gmp_randclear" gmp_randclear :: Ptr GMPRandState ->↙
    ↳  IO ()
-- ** Random State Seeding
foreign import ccall safe "__gmp_randseed" gmp_randseed :: Ptr GMPRandState -> ↙
    ↳ SrcPtr MPZ -> IO ()
foreign import ccall safe "__gmp_randseed_ui" gmp_randseed_ui :: Ptr ↙
    ↳ GMPRandState -> CULong -> IO ()
-- ** Random State Miscellaneous
foreign import ccall safe "__gmp_urandomb_ui" gmp_urandomb_ui :: Ptr ↙
    ↳ GMPRandState -> CULong -> IO CULong
foreign import ccall safe "__gmp_urandomm_ui" gmp_urandomm_ui :: Ptr ↙
    ↳ GMPRandState -> CULong -> IO CULong

-- * Low-level Functions
foreign import ccall safe "__gmpn_add_n" mpn_add_n :: Ptr MPLimb -> SrcPtr ↙
    ↳ MPLimb -> SrcPtr MPLimb -> MPSize -> IO MPLimb
foreign import ccall safe "__gmpn_addmul_1" mpn_addmul_1 :: Ptr MPLimb -> SrcPtr↙
    ↳  MPLimb -> MPSize -> MPLimb -> IO MPLimb
foreign import ccall safe "__gmpn_divexact_1" mpn_divexact_1 :: Ptr MPLimb -> ↙
    ↳ SrcPtr MPLimb -> MPSize -> MPLimb -> IO ()
foreign import ccall safe "__gmpn_divexact_by3c" mpn_divexact_by3c :: Ptr MPLimb↙
    ↳  -> SrcPtr MPLimb -> MPSize -> MPLimb -> IO MPLimb
foreign import ccall safe "__gmpn_divrem" mpn_divrem :: Ptr MPLimb -> MPSize -> ↙
    ↳ Ptr MPLimb -> MPSize -> SrcPtr MPLimb -> MPSize -> IO MPLimb
foreign import ccall safe "__gmpn_divrem_1" mpn_divrem_1 :: Ptr MPLimb -> MPSize↙
    ↳  -> SrcPtr MPLimb -> MPSize -> MPLimb -> IO MPLimb
foreign import ccall safe "__gmpn_divrem_2" mpn_divrem_2 :: Ptr MPLimb -> MPSize↙
```

```
    ↪ -> Ptr MPLimb -> MPSize -> SrcPtr MPLimb -> IO MPLimb
foreign import ccall safe "__gmpn_div_qr_1" mpn_div_qr_1 :: Ptr MPLimb -> Ptr ↙
    ↪ MPLimb -> SrcPtr MPLimb -> MPSize -> MPLimb -> IO MPLimb
foreign import ccall safe "__gmpn_div_qr_2" mpn_div_qr_2 :: Ptr MPLimb -> Ptr ↙
    ↪ MPLimb -> SrcPtr MPLimb -> MPSize -> SrcPtr MPLimb -> IO MPLimb
foreign import ccall safe "__gmpn_gcd" mpn_gcd :: Ptr MPLimb -> Ptr MPLimb -> ↙
    ↪ MPSize -> Ptr MPLimb -> MPSize -> IO MPSize
foreign import ccall safe "__gmpn_gcd_1" mpn_gcd_1 :: SrcPtr MPLimb -> MPSize ->↙
    ↪ MPLimb -> IO MPLimb
foreign import ccall safe "__gmpn_gcdext_1" mpn_gcdext_1 :: Ptr MPLimbSigned -> ↙
    ↪ Ptr MPLimbSigned -> MPLimb -> MPLimb -> IO MPLimb
foreign import ccall safe "__gmpn_gcdext" mpn_gcdext :: Ptr MPLimb -> Ptr MPLimb↙
    ↪ -> Ptr MPSize -> Ptr MPLimb -> MPSize -> Ptr MPLimb -> MPSize -> IO ↙
    ↪ MPSize
foreign import ccall safe "__gmpn_get_str" mpn_get_str :: Ptr CUChar -> CInt -> ↙
    ↪ Ptr MPLimb -> MPSize -> IO CSize
foreign import ccall safe "__gmpn_hamdist" mpn_hamdist :: SrcPtr MPLimb -> ↙
    ↪ SrcPtr MPLimb -> MPSize -> IO MPBitCnt
foreign import ccall safe "__gmpn_lshift" mpn_lshift :: Ptr MPLimb -> SrcPtr ↙
    ↪ MPLimb -> MPSize -> CUInt -> IO MPLimb
foreign import ccall safe "__gmpn_mod_1" mpn_mod_1 :: SrcPtr MPLimb -> MPSize ->↙
    ↪ MPLimb -> IO MPLimb
foreign import ccall safe "__gmpn_mul" mpn_mul :: Ptr MPLimb -> SrcPtr MPLimb ->↙
    ↪ MPSize -> SrcPtr MPLimb -> MPSize -> IO MPLimb
foreign import ccall safe "__gmpn_mul_1" mpn_mul_1 :: Ptr MPLimb -> SrcPtr ↙
    ↪ MPLimb -> MPSize -> MPLimb -> IO MPLimb
foreign import ccall safe "__gmpn_mul_n" mpn_mul_n :: Ptr MPLimb -> SrcPtr ↙
    ↪ MPLimb -> SrcPtr MPLimb -> MPSize -> IO ()
foreign import ccall safe "__gmpn_sqr" mpn_sqr :: Ptr MPLimb -> SrcPtr MPLimb ->↙
    ↪ MPSize -> IO ()
foreign import ccall safe "__gmpn_com" mpn_com :: Ptr MPLimb -> SrcPtr MPLimb ->↙
    ↪ MPSize -> IO ()
foreign import ccall safe "__gmpn_perfect_square_p" mpn_perfect_square_p :: ↙
    ↪ SrcPtr MPLimb -> MPSize -> IO CInt
foreign import ccall safe "__gmpn_perfect_power_p" mpn_perfect_power_p :: SrcPtr↙
    ↪ MPLimb -> MPSize -> IO CInt
foreign import ccall safe "__gmpn_popcount" mpn_popcount :: SrcPtr MPLimb -> ↙
    ↪ MPSize -> IO MPBitCnt
foreign import ccall safe "__gmpn_pow_1" mpn_pow_1 :: Ptr MPLimb -> SrcPtr ↙
    ↪ MPLimb -> MPSize -> MPLimb -> Ptr MPLimb -> IO MPSize
foreign import ccall safe "__gmpn_preinv_mod_1" mpn_preinv_mod_1 :: SrcPtr ↙
    ↪ MPLimb -> MPSize -> MPLimb -> MPLimb -> IO MPLimb
foreign import ccall safe "__gmpn_random" mpn_random :: Ptr MPLimb -> MPSize -> ↙
    ↪ IO ()
foreign import ccall safe "__gmpn_random2" mpn_random2 :: Ptr MPLimb -> MPSize ↙
    ↪ -> IO ()
foreign import ccall safe "__gmpn_rshift" mpn_rshift :: Ptr MPLimb -> SrcPtr ↙
    ↪ MPLimb -> MPSize -> CUInt -> IO MPLimb
foreign import ccall safe "__gmpn_scan0" mpn_scan0 :: SrcPtr MPLimb -> MPBitCnt ↙
    ↪ -> IO MPBitCnt
foreign import ccall safe "__gmpn_scan1" mpn_scan1 :: SrcPtr MPLimb -> MPBitCnt ↙
    ↪ -> IO MPBitCnt
foreign import ccall safe "__gmpn_set_str" mpn_set_str :: Ptr MPLimb -> SrcPtr ↙
    ↪ CChar -> CSize -> CInt -> IO MPSize
foreign import ccall safe "__gmpn_sizeinbase" mpn_sizeinbase :: SrcPtr MPLimb ->↙
    ↪ MPSize -> CInt -> IO CSize
foreign import ccall safe "__gmpn_sqrtrem" mpn_sqrtrem :: Ptr MPLimb -> Ptr ↙
```

```
    ↳ MPLimb -> SrcPtr MPLimb -> MPSize -> IO MPSize
foreign import ccall safe "__gmpn_sub_n" mpn_sub_n :: Ptr MPLimb -> SrcPtr ↙
    ↳ MPLimb -> SrcPtr MPLimb -> MPSize -> IO MPLimb
foreign import ccall safe "__gmpn_submul_1" mpn_submul_1 :: Ptr MPLimb -> SrcPtr↙
    ↳  MPLimb -> MPSize -> MPLimb -> IO MPLimb
foreign import ccall safe "__gmpn_tdiv_qr" mpn_tdiv_qr :: Ptr MPLimb -> Ptr ↙
    ↳ MPLimb -> MPSize -> SrcPtr MPLimb -> MPSize -> SrcPtr MPLimb -> MPSize -> ↙
    ↳ IO ()
foreign import ccall safe "__gmpn_and_n" mpn_and_n :: Ptr MPLimb -> SrcPtr ↙
    ↳ MPLimb -> SrcPtr MPLimb -> MPSize -> IO ()
foreign import ccall safe "__gmpn_andn_n" mpn_andn_n :: Ptr MPLimb -> SrcPtr ↙
    ↳ MPLimb -> SrcPtr MPLimb -> MPSize -> IO ()
foreign import ccall safe "__gmpn_nand_n" mpn_nand_n :: Ptr MPLimb -> SrcPtr ↙
    ↳ MPLimb -> SrcPtr MPLimb -> MPSize -> IO ()
foreign import ccall safe "__gmpn_ior_n" mpn_ior_n :: Ptr MPLimb -> SrcPtr ↙
    ↳ MPLimb -> SrcPtr MPLimb -> MPSize -> IO ()
foreign import ccall safe "__gmpn_iorn_n" mpn_iorn_n :: Ptr MPLimb -> SrcPtr ↙
    ↳ MPLimb -> SrcPtr MPLimb -> MPSize -> IO ()
foreign import ccall safe "__gmpn_nior_n" mpn_nior_n :: Ptr MPLimb -> SrcPtr ↙
    ↳ MPLimb -> SrcPtr MPLimb -> MPSize -> IO ()
foreign import ccall safe "__gmpn_xor_n" mpn_xor_n :: Ptr MPLimb -> SrcPtr ↙
    ↳ MPLimb -> SrcPtr MPLimb -> MPSize -> IO ()
foreign import ccall safe "__gmpn_xnor_n" mpn_xnor_n :: Ptr MPLimb -> SrcPtr ↙
    ↳ MPLimb -> SrcPtr MPLimb -> MPSize -> IO ()
foreign import ccall safe "__gmpn_copyi" mpn_copyi :: Ptr MPLimb -> SrcPtr ↙
    ↳ MPLimb -> MPSize -> IO ()
foreign import ccall safe "__gmpn_copyd" mpn_copyd :: Ptr MPLimb -> SrcPtr ↙
    ↳ MPLimb -> MPSize -> IO ()
foreign import ccall safe "__gmpn_zero" mpn_zero :: Ptr MPLimb -> MPSize -> IO ↙
    ↳ ()
foreign import ccall safe "__gmpn_cnd_add_n" mpn_cnd_add_n :: MPLimb -> Ptr ↙
    ↳ MPLimb -> SrcPtr MPLimb -> SrcPtr MPLimb -> MPSize -> IO MPLimb
foreign import ccall safe "__gmpn_cnd_sub_n" mpn_cnd_sub_n :: MPLimb -> Ptr ↙
    ↳ MPLimb -> SrcPtr MPLimb -> SrcPtr MPLimb -> MPSize -> IO MPLimb
foreign import ccall safe "__gmpn_sec_add_1" mpn_sec_add_1 :: Ptr MPLimb -> ↙
    ↳ SrcPtr MPLimb -> MPSize -> MPLimb -> Ptr MPLimb -> IO MPLimb
foreign import ccall safe "__gmpn_sec_add_1_itch" mpn_sec_add_1_itch :: MPSize ↙
    ↳ -> IO MPSize
foreign import ccall safe "__gmpn_sec_sub_1" mpn_sec_sub_1 :: Ptr MPLimb -> ↙
    ↳ SrcPtr MPLimb -> MPSize -> MPLimb -> Ptr MPLimb -> IO MPLimb
foreign import ccall safe "__gmpn_sec_sub_1_itch" mpn_sec_sub_1_itch :: MPSize ↙
    ↳ -> IO MPSize
foreign import ccall safe "__gmpn_cnd_swap" mpn_cnd_swap :: MPLimb -> ↙
    ↳ VolatilePtr MPLimb -> VolatilePtr MPLimb -> MPSize -> IO ()
foreign import ccall safe "__gmpn_sec_mul" mpn_sec_mul :: Ptr MPLimb -> SrcPtr ↙
    ↳ MPLimb -> MPSize -> SrcPtr MPLimb -> MPSize -> Ptr MPLimb -> IO ()
foreign import ccall safe "__gmpn_sec_mul_itch" mpn_sec_mul_itch :: MPSize -> ↙
    ↳ MPSize -> IO MPSize
foreign import ccall safe "__gmpn_sec_sqr" mpn_sec_sqr :: Ptr MPLimb -> SrcPtr ↙
    ↳ MPLimb -> MPSize -> Ptr MPLimb -> IO ()
foreign import ccall safe "__gmpn_sec_sqr_itch" mpn_sec_sqr_itch :: MPSize -> IO↙
    ↳  MPSize
foreign import ccall safe "__gmpn_sec_powm" mpn_sec_powm :: Ptr MPLimb -> SrcPtr↙
    ↳  MPLimb -> MPSize -> SrcPtr MPLimb -> MPBitCnt -> SrcPtr MPLimb -> MPSize ↙
    ↳ -> Ptr MPLimb -> IO ()
foreign import ccall safe "__gmpn_sec_powm_itch" mpn_sec_powm_itch :: MPSize -> ↙
    ↳ MPBitCnt -> MPSize -> IO MPSize
```

```
foreign import ccall safe "__gmpn_sec_tabselect" mpn_sec_tabselect :: ↙
    ↳ VolatilePtr MPLimb -> VolatileSrcPtr MPLimb -> MPSize -> MPSize -> MPSize ↙
    ↳ -> IO ()
foreign import ccall safe "__gmpn_sec_div_qr" mpn_sec_div_qr :: Ptr MPLimb -> ↙
    ↳ Ptr MPLimb -> MPSize -> SrcPtr MPLimb -> MPSize -> Ptr MPLimb -> IO MPLimb
foreign import ccall safe "__gmpn_sec_div_qr_itch" mpn_sec_div_qr_itch :: MPSize↙
    ↳  -> MPSize -> IO MPSize
foreign import ccall safe "__gmpn_sec_div_r" mpn_sec_div_r :: Ptr MPLimb -> ↙
    ↳ MPSize -> SrcPtr MPLimb -> MPSize -> Ptr MPLimb -> IO ()
foreign import ccall safe "__gmpn_sec_div_r_itch" mpn_sec_div_r_itch :: MPSize ↙
    ↳ -> MPSize -> IO MPSize
foreign import ccall safe "__gmpn_sec_invert" mpn_sec_invert :: Ptr MPLimb -> ↙
    ↳ Ptr MPLimb -> SrcPtr MPLimb -> MPSize -> MPBitCnt -> Ptr MPLimb -> IO CInt
foreign import ccall safe "__gmpn_sec_invert_itch" mpn_sec_invert_itch :: MPSize↙
    ↳  -> IO MPSize

{-
foreign import ccall safe "__gmpz_dump" mpz_dump :: SrcPtr MPZ -> IO ()
foreign import ccall safe "__gmpz_millerrabin" mpz_millerrabin :: SrcPtr MPZ -> ↙
    ↳ CInt -> IO CInt
foreign import ccall safe "__gmpf_dump" mpf_dump :: SrcPtr MPF -> IO ()
foreign import ccall safe "__gmpf_eq" mpf_eq :: SrcPtr MPF -> SrcPtr MPF -> ↙
    ↳ MPBitCnt -> IO CInt
foreign import ccall safe "__gmpf_size" mpf_size :: SrcPtr MPF -> IO CSize
-}
```

10 src/Numeric/GMP/Raw/Unsafe.hs

```
{-# LANGUAGE ForeignFunctionInterface #-}
-- | Raw GMP foreign bindings, imported unsafe.
module Numeric.GMP.Raw.Unsafe where

import Foreign.Ptr (Ptr)
import Foreign.C.Types
import Numeric.GMP.Types

-- * Types for Documentation

type SrcPtr = Ptr
type VolatilePtr = Ptr
type VolatileSrcPtr = Ptr

-- * Integer Functions
-- ** Initialization Functions
foreign import ccall unsafe "__gmpz_init" mpz_init :: Ptr MPZ -> IO ()
foreign import ccall unsafe "__gmpz_init2" mpz_init2 :: Ptr MPZ -> MPBitCnt -> ↙
    ↳ IO ()
foreign import ccall unsafe "__gmpz_clear" mpz_clear :: Ptr MPZ -> IO ()
foreign import ccall unsafe "__gmpz_realloc2" mpz_realloc2 :: Ptr MPZ -> ↙
    ↳ MPBitCnt -> IO ()
-- ** Assignment Functions
foreign import ccall unsafe "__gmpz_set" mpz_set :: Ptr MPZ -> SrcPtr MPZ -> IO ↙
    ↳ ()
foreign import ccall unsafe "__gmpz_set_ui" mpz_set_ui :: Ptr MPZ -> CULong -> ↙
    ↳ IO ()
foreign import ccall unsafe "__gmpz_set_si" mpz_set_si :: Ptr MPZ -> CLong -> IO↙
    ↳  ()
```

```
25  foreign import ccall unsafe "__gmpz_set_d" mpz_set_d :: Ptr MPZ -> CDouble -> IO ()
    foreign import ccall unsafe "__gmpz_set_q" mpz_set_q :: Ptr MPZ -> SrcPtr MPQ -> IO ()
    foreign import ccall unsafe "__gmpz_set_f" mpz_set_f :: Ptr MPZ -> SrcPtr MPF -> IO ()
    foreign import ccall unsafe "__gmpz_set_str" mpz_set_str :: Ptr MPZ -> SrcPtr CChar -> CInt -> IO CInt
    foreign import ccall unsafe "__gmpz_swap" mpz_swap :: Ptr MPZ -> Ptr MPZ -> IO ()
30  -- ** Combined Initialization and Assignment Functions
    foreign import ccall unsafe "__gmpz_init_set" mpz_init_set :: Ptr MPZ -> SrcPtr MPZ -> IO ()
    foreign import ccall unsafe "__gmpz_init_set_ui" mpz_init_set_ui :: Ptr MPZ -> CULong -> IO ()
    foreign import ccall unsafe "__gmpz_init_set_si" mpz_init_set_si :: Ptr MPZ -> CLong -> IO ()
    foreign import ccall unsafe "__gmpz_init_set_d" mpz_init_set_d :: Ptr MPZ -> CDouble -> IO ()
35  foreign import ccall unsafe "__gmpz_init_set_str" mpz_init_set_str :: Ptr MPZ -> SrcPtr CChar -> CInt -> IO CInt
    -- ** Conversion Functions
    foreign import ccall unsafe "__gmpz_get_ui" mpz_get_ui :: SrcPtr MPZ -> IO CLong
    foreign import ccall unsafe "__gmpz_get_si" mpz_get_si :: SrcPtr MPZ -> IO CLong
    foreign import ccall unsafe "__gmpz_get_d" mpz_get_d :: SrcPtr MPZ -> IO CDouble
40  foreign import ccall unsafe "__gmpz_get_d_2exp" mpz_get_d_2exp :: Ptr CLong -> SrcPtr MPZ -> IO CDouble
    foreign import ccall unsafe "__gmpz_get_str" mpz_get_str :: Ptr CChar -> CInt -> SrcPtr MPZ -> IO CChar
    -- ** Arithmetic Functions
    foreign import ccall unsafe "__gmpz_add" mpz_add :: Ptr MPZ -> SrcPtr MPZ -> SrcPtr MPZ -> IO ()
    foreign import ccall unsafe "__gmpz_add_ui" mpz_add_ui :: Ptr MPZ -> SrcPtr MPZ -> CULong -> IO ()
45  foreign import ccall unsafe "__gmpz_sub" mpz_sub :: Ptr MPZ -> SrcPtr MPZ -> SrcPtr MPZ -> IO ()
    foreign import ccall unsafe "__gmpz_sub_ui" mpz_sub_ui :: Ptr MPZ -> SrcPtr MPZ -> CULong -> IO ()
    foreign import ccall unsafe "__gmpz_ui_sub" mpz_ui_sub :: Ptr MPZ -> CULong -> SrcPtr MPZ -> IO ()
    foreign import ccall unsafe "__gmpz_mul" mpz_mul :: Ptr MPZ -> SrcPtr MPZ -> SrcPtr MPZ -> IO ()
    foreign import ccall unsafe "__gmpz_mul_si" mpz_mul_si :: Ptr MPZ -> SrcPtr MPZ -> CLong -> IO ()
50  foreign import ccall unsafe "__gmpz_mul_ui" mpz_mul_ui :: Ptr MPZ -> SrcPtr MPZ -> CULong -> IO ()
    foreign import ccall unsafe "__gmpz_addmul" mpz_addmul :: Ptr MPZ -> SrcPtr MPZ -> SrcPtr MPZ -> IO ()
    foreign import ccall unsafe "__gmpz_addmul_ui" mpz_addmul_ui :: Ptr MPZ -> SrcPtr MPZ -> CULong -> IO ()
    foreign import ccall unsafe "__gmpz_submul" mpz_submul :: Ptr MPZ -> SrcPtr MPZ -> SrcPtr MPZ -> IO ()
    foreign import ccall unsafe "__gmpz_submul_ui" mpz_submul_ui :: Ptr MPZ -> SrcPtr MPZ -> CULong -> IO ()
55  foreign import ccall unsafe "__gmpz_mul_2exp" mpz_mul_2exp :: Ptr MPZ -> SrcPtr MPZ -> MPBitCnt -> IO ()
    -- mpz_neg
```

```
-- mpz_abs
-- ** Division Functions
foreign import ccall unsafe "__gmpz_cdiv_q" mpz_cdiv_q :: Ptr MPZ -> SrcPtr MPZ ↙
    ↳ -> SrcPtr MPZ -> IO ()
foreign import ccall unsafe "__gmpz_cdiv_r" mpz_cdiv_r :: Ptr MPZ -> SrcPtr MPZ ↙
    ↳ -> SrcPtr MPZ -> IO ()
foreign import ccall unsafe "__gmpz_cdiv_qr" mpz_cdiv_qr :: Ptr MPZ -> Ptr MPZ ↙
    ↳ -> SrcPtr MPZ -> SrcPtr MPZ -> IO ()
foreign import ccall unsafe "__gmpz_cdiv_q_ui" mpz_cdiv_q_ui :: Ptr MPZ -> ↙
    ↳ SrcPtr MPZ -> CULong -> IO CULong
foreign import ccall unsafe "__gmpz_cdiv_r_ui" mpz_cdiv_r_ui :: Ptr MPZ -> ↙
    ↳ SrcPtr MPZ -> CULong -> IO CULong
foreign import ccall unsafe "__gmpz_cdiv_qr_ui" mpz_cdiv_qr_ui :: Ptr MPZ -> Ptr↙
    ↳  MPZ -> SrcPtr MPZ -> CULong -> IO CULong
foreign import ccall unsafe "__gmpz_cdiv_ui" mpz_cdiv_ui :: SrcPtr MPZ -> CULong↙
    ↳  -> IO CULong
foreign import ccall unsafe "__gmpz_cdiv_q_2exp" mpz_cdiv_q_2exp :: Ptr MPZ -> ↙
    ↳ SrcPtr MPZ -> MPBitCnt -> IO ()
foreign import ccall unsafe "__gmpz_cdiv_r_2exp" mpz_cdiv_r_2exp :: Ptr MPZ -> ↙
    ↳ SrcPtr MPZ -> MPBitCnt -> IO ()
foreign import ccall unsafe "__gmpz_fdiv_q" mpz_fdiv_q :: Ptr MPZ -> SrcPtr MPZ ↙
    ↳ -> SrcPtr MPZ -> IO ()
foreign import ccall unsafe "__gmpz_fdiv_r" mpz_fdiv_r :: Ptr MPZ -> SrcPtr MPZ ↙
    ↳ -> SrcPtr MPZ -> IO ()
foreign import ccall unsafe "__gmpz_fdiv_qr" mpz_fdiv_qr :: Ptr MPZ -> Ptr MPZ ↙
    ↳ -> SrcPtr MPZ -> SrcPtr MPZ -> IO ()
foreign import ccall unsafe "__gmpz_fdiv_q_ui" mpz_fdiv_q_ui :: Ptr MPZ -> ↙
    ↳ SrcPtr MPZ -> CULong -> IO CULong
foreign import ccall unsafe "__gmpz_fdiv_r_ui" mpz_fdiv_r_ui :: Ptr MPZ -> ↙
    ↳ SrcPtr MPZ -> CULong -> IO CULong
foreign import ccall unsafe "__gmpz_fdiv_qr_ui" mpz_fdiv_qr_ui :: Ptr MPZ -> Ptr↙
    ↳  MPZ -> SrcPtr MPZ -> CULong -> IO CULong
foreign import ccall unsafe "__gmpz_fdiv_ui" mpz_fdiv_ui :: SrcPtr MPZ -> CULong↙
    ↳  -> IO CULong
foreign import ccall unsafe "__gmpz_fdiv_r_2exp" mpz_fdiv_r_2exp :: Ptr MPZ -> ↙
    ↳ SrcPtr MPZ -> MPBitCnt -> IO ()
foreign import ccall unsafe "__gmpz_fdiv_q_2exp" mpz_fdiv_q_2exp :: Ptr MPZ -> ↙
    ↳ SrcPtr MPZ -> MPBitCnt -> IO ()
foreign import ccall unsafe "__gmpz_tdiv_q" mpz_tdiv_q :: Ptr MPZ -> SrcPtr MPZ ↙
    ↳ -> SrcPtr MPZ -> IO ()
foreign import ccall unsafe "__gmpz_tdiv_r" mpz_tdiv_r :: Ptr MPZ -> SrcPtr MPZ ↙
    ↳ -> SrcPtr MPZ -> IO ()
foreign import ccall unsafe "__gmpz_tdiv_qr" mpz_tdiv_qr :: Ptr MPZ -> Ptr MPZ ↙
    ↳ -> SrcPtr MPZ -> SrcPtr MPZ -> IO ()
foreign import ccall unsafe "__gmpz_tdiv_q_ui" mpz_tdiv_q_ui :: Ptr MPZ -> ↙
    ↳ SrcPtr MPZ -> CULong -> IO CULong
foreign import ccall unsafe "__gmpz_tdiv_r_ui" mpz_tdiv_r_ui :: Ptr MPZ -> ↙
    ↳ SrcPtr MPZ -> CULong -> IO CULong
foreign import ccall unsafe "__gmpz_tdiv_qr_ui" mpz_tdiv_qr_ui :: Ptr MPZ -> Ptr↙
    ↳  MPZ -> SrcPtr MPZ -> CULong -> IO CULong
foreign import ccall unsafe "__gmpz_tdiv_ui" mpz_tdiv_ui :: SrcPtr MPZ -> CULong↙
    ↳  -> IO CULong
foreign import ccall unsafe "__gmpz_tdiv_q_2exp" mpz_tdiv_q_2exp :: Ptr MPZ -> ↙
    ↳ SrcPtr MPZ -> MPBitCnt -> IO ()
foreign import ccall unsafe "__gmpz_tdiv_r_2exp" mpz_tdiv_r_2exp :: Ptr MPZ -> ↙
    ↳ SrcPtr MPZ -> MPBitCnt -> IO ()
foreign import ccall unsafe "__gmpz_mod" mpz_mod :: Ptr MPZ -> SrcPtr MPZ -> ↙
```

```
    ↳ SrcPtr MPZ -> IO ()
-- mpz_mod_ui
foreign import ccall unsafe "__gmpz_divexact" mpz_divexact :: Ptr MPZ -> SrcPtr ↙
    ↳ MPZ -> SrcPtr MPZ -> IO ()
foreign import ccall unsafe "__gmpz_divexact_ui" mpz_divexact_ui :: Ptr MPZ -> ↙
    ↳ SrcPtr MPZ -> CULong -> IO ()
foreign import ccall unsafe "__gmpz_divisible_p" mpz_divisible_p :: SrcPtr MPZ ↙
    ↳ -> SrcPtr MPZ -> IO CInt
foreign import ccall unsafe "__gmpz_divisible_ui_p" mpz_divisible_ui_p :: SrcPtr↙
    ↳  MPZ -> CULong -> IO CInt
foreign import ccall unsafe "__gmpz_divisible_2exp_p" mpz_divisible_2exp_p :: ↙
    ↳ SrcPtr MPZ -> MPBitCnt -> IO CInt
foreign import ccall unsafe "__gmpz_congruent_p" mpz_congruent_p :: SrcPtr MPZ ↙
    ↳ -> SrcPtr MPZ -> SrcPtr MPZ -> IO CInt
foreign import ccall unsafe "__gmpz_congruent_ui_p" mpz_congruent_ui_p :: SrcPtr↙
    ↳  MPZ -> CULong -> CULong -> IO CInt
foreign import ccall unsafe "__gmpz_congruent_2exp_p" mpz_congruent_2exp_p :: ↙
    ↳ SrcPtr MPZ -> SrcPtr MPZ -> MPBitCnt -> IO CInt
-- ** Exponentiation Functions
foreign import ccall unsafe "__gmpz_powm" mpz_powm :: Ptr MPZ -> SrcPtr MPZ -> ↙
    ↳ SrcPtr MPZ -> SrcPtr MPZ -> IO ()
foreign import ccall unsafe "__gmpz_powm_ui" mpz_powm_ui :: Ptr MPZ -> SrcPtr ↙
    ↳ MPZ -> CULong -> SrcPtr MPZ -> IO ()
foreign import ccall unsafe "__gmpz_powm_sec" mpz_powm_sec :: Ptr MPZ -> SrcPtr ↙
    ↳ MPZ -> SrcPtr MPZ -> SrcPtr MPZ -> IO ()
foreign import ccall unsafe "__gmpz_pow_ui" mpz_pow_ui :: Ptr MPZ -> SrcPtr MPZ ↙
    ↳ -> CULong -> IO ()
foreign import ccall unsafe "__gmpz_ui_pow_ui" mpz_ui_pow_ui :: Ptr MPZ -> ↙
    ↳ CULong -> CULong -> IO ()
-- ** Root Extraction Functions
foreign import ccall unsafe "__gmpz_root" mpz_root :: Ptr MPZ -> SrcPtr MPZ -> ↙
    ↳ CULong -> IO CInt
foreign import ccall unsafe "__gmpz_rootrem" mpz_rootrem :: Ptr MPZ -> Ptr MPZ ↙
    ↳ -> SrcPtr MPZ -> CULong -> IO ()
foreign import ccall unsafe "__gmpz_sqrt" mpz_sqrt :: Ptr MPZ -> SrcPtr MPZ -> ↙
    ↳ IO ()
foreign import ccall unsafe "__gmpz_sqrtrem" mpz_sqrtrem :: Ptr MPZ -> Ptr MPZ ↙
    ↳ -> SrcPtr MPZ -> IO ()
foreign import ccall unsafe "__gmpz_perfect_power_p" mpz_perfect_power_p :: ↙
    ↳ SrcPtr MPZ -> IO CInt
-- mpz_perfect_square_p
-- ** Number Theoretic Functions
foreign import ccall unsafe "__gmpz_probab_prime_p" mpz_probab_prime_p :: SrcPtr↙
    ↳  MPZ -> CInt -> IO CInt
foreign import ccall unsafe "__gmpz_nextprime" mpz_nextprime :: Ptr MPZ -> ↙
    ↳ SrcPtr MPZ -> IO ()
foreign import ccall unsafe "__gmpz_gcd" mpz_gcd :: Ptr MPZ -> SrcPtr MPZ -> ↙
    ↳ SrcPtr MPZ -> IO ()
foreign import ccall unsafe "__gmpz_gcd_ui" mpz_gcd_ui :: Ptr MPZ -> SrcPtr MPZ ↙
    ↳ -> CULong -> IO CULong
foreign import ccall unsafe "__gmpz_gcdext" mpz_gcdext :: Ptr MPZ -> Ptr MPZ -> ↙
    ↳ Ptr MPZ -> SrcPtr MPZ -> SrcPtr MPZ -> IO ()
foreign import ccall unsafe "__gmpz_lcm" mpz_lcm :: Ptr MPZ -> SrcPtr MPZ -> ↙
    ↳ SrcPtr MPZ -> IO ()
foreign import ccall unsafe "__gmpz_lcm_ui" mpz_lcm_ui :: Ptr MPZ -> SrcPtr MPZ ↙
    ↳ -> CULong -> IO ()
foreign import ccall unsafe "__gmpz_invert" mpz_invert :: Ptr MPZ -> SrcPtr MPZ ↙
```

```
    ↳ -> SrcPtr MPZ -> IO CInt
foreign import ccall unsafe "__gmpz_jacobi" mpz_jacobi :: SrcPtr MPZ -> SrcPtr ↙
    ↳ MPZ -> IO CInt
-- mpz_legendre
-- mpz_kronecker
foreign import ccall unsafe "__gmpz_kronecker_si" mpz_kronecker_si :: SrcPtr MPZ↙
    ↳  -> CLong -> IO CInt
foreign import ccall unsafe "__gmpz_kronecker_ui" mpz_kronecker_ui :: SrcPtr MPZ↙
    ↳  -> CULong -> IO CInt
foreign import ccall unsafe "__gmpz_si_kronecker" mpz_si_kronecker :: CLong -> ↙
    ↳ SrcPtr MPZ -> IO CInt
foreign import ccall unsafe "__gmpz_ui_kronecker" mpz_ui_kronecker :: CULong -> ↙
    ↳ SrcPtr MPZ -> IO CInt
foreign import ccall unsafe "__gmpz_remove" mpz_remove :: Ptr MPZ -> SrcPtr MPZ ↙
    ↳ -> SrcPtr MPZ -> IO MPBitCnt
foreign import ccall unsafe "__gmpz_fac_ui" mpz_fac_ui :: Ptr MPZ -> CULong -> ↙
    ↳ IO ()
foreign import ccall unsafe "__gmpz_2fac_ui" mpz_2fac_ui :: Ptr MPZ -> CULong ->↙
    ↳  IO ()
foreign import ccall unsafe "__gmpz_mfac_uiui" mpz_mfac_uiui :: Ptr MPZ -> ↙
    ↳ CULong -> CULong -> IO ()
foreign import ccall unsafe "__gmpz_primorial_ui" mpz_primorial_ui :: Ptr MPZ ->↙
    ↳  CULong -> IO ()
foreign import ccall unsafe "__gmpz_bin_ui" mpz_bin_ui :: Ptr MPZ -> SrcPtr MPZ ↙
    ↳ -> CULong -> IO ()
foreign import ccall unsafe "__gmpz_bin_uiui" mpz_bin_uiui :: Ptr MPZ -> CULong ↙
    ↳ -> CULong -> IO ()
foreign import ccall unsafe "__gmpz_fib_ui" mpz_fib_ui :: Ptr MPZ -> CULong -> ↙
    ↳ IO ()
foreign import ccall unsafe "__gmpz_fib2_ui" mpz_fib2_ui :: Ptr MPZ -> Ptr MPZ ↙
    ↳ -> CULong -> IO ()
foreign import ccall unsafe "__gmpz_lucnum_ui" mpz_lucnum_ui :: Ptr MPZ -> ↙
    ↳ CULong -> IO ()
foreign import ccall unsafe "__gmpz_lucnum2_ui" mpz_lucnum2_ui :: Ptr MPZ -> Ptr↙
    ↳  MPZ -> CULong -> IO ()
-- ** Comparison Functions
foreign import ccall unsafe "__gmpz_cmp" mpz_cmp :: SrcPtr MPZ -> SrcPtr MPZ -> ↙
    ↳ IO CInt
foreign import ccall unsafe "__gmpz_cmp_d" mpz_cmp_d :: SrcPtr MPZ -> CDouble ->↙
    ↳  IO CInt
foreign import ccall unsafe "__gmpz_cmp_si" mpz_cmp_si :: SrcPtr MPZ -> CLong ->↙
    ↳  IO CInt
foreign import ccall unsafe "__gmpz_cmp_ui" mpz_cmp_ui :: SrcPtr MPZ -> CULong ↙
    ↳ -> IO CInt
foreign import ccall unsafe "__gmpz_cmpabs" mpz_cmpabs :: SrcPtr MPZ -> SrcPtr ↙
    ↳ MPZ -> IO CInt
foreign import ccall unsafe "__gmpz_cmpabs_d" mpz_cmpabs_d :: SrcPtr MPZ -> ↙
    ↳ CDouble -> IO CInt
foreign import ccall unsafe "__gmpz_cmpabs_ui" mpz_cmpabs_ui :: SrcPtr MPZ -> ↙
    ↳ CULong -> IO CInt
-- mpz_sgn
-- ** Logical and Bit Manipulation Functions
foreign import ccall unsafe "__gmpz_and" mpz_and :: Ptr MPZ -> SrcPtr MPZ -> ↙
    ↳ SrcPtr MPZ -> IO ()
foreign import ccall unsafe "__gmpz_ior" mpz_ior :: Ptr MPZ -> SrcPtr MPZ -> ↙
    ↳ SrcPtr MPZ -> IO ()
foreign import ccall unsafe "__gmpz_xor" mpz_xor :: Ptr MPZ -> SrcPtr MPZ -> ↙
```

```
    ↳ SrcPtr MPZ -> IO ()
foreign import ccall unsafe "__gmpz_com" mpz_com :: Ptr MPZ -> SrcPtr MPZ -> IO ↙
    ↳ ()
-- mpz_popcount
foreign import ccall unsafe "__gmpz_hamdist" mpz_hamdist :: SrcPtr MPZ -> SrcPtr↙
    ↳ MPZ -> IO MPBitCnt
foreign import ccall unsafe "__gmpz_scan0" mpz_scan0 :: SrcPtr MPZ -> MPBitCnt ↙
    ↳ -> IO MPBitCnt
foreign import ccall unsafe "__gmpz_scan1" mpz_scan1 :: SrcPtr MPZ -> MPBitCnt ↙
    ↳ -> IO MPBitCnt
foreign import ccall unsafe "__gmpz_setbit" mpz_setbit :: Ptr MPZ -> MPBitCnt ->↙
    ↳ IO ()
foreign import ccall unsafe "__gmpz_clrbit" mpz_clrbit :: Ptr MPZ -> MPBitCnt ->↙
    ↳ IO ()
foreign import ccall unsafe "__gmpz_combit" mpz_combit :: Ptr MPZ -> MPBitCnt ->↙
    ↳ IO ()
foreign import ccall unsafe "__gmpz_tstbit" mpz_tstbit :: SrcPtr MPZ -> MPBitCnt↙
    ↳ -> IO CInt
-- ** Input and Output Functions
foreign import ccall unsafe "__gmpz_out_str" mpz_out_str :: Ptr CFile -> CInt ->↙
    ↳ SrcPtr MPZ -> IO CSize
foreign import ccall unsafe "__gmpz_inp_str" mpz_inp_str :: Ptr MPZ -> Ptr CFile↙
    ↳ -> CInt -> IO CSize
foreign import ccall unsafe "__gmpz_out_raw" mpz_out_raw :: Ptr CFile -> SrcPtr ↙
    ↳ MPZ -> IO CSize
foreign import ccall unsafe "__gmpz_inp_raw" mpz_inp_raw :: Ptr MPZ -> Ptr CFile↙
    ↳ -> IO CSize
-- ** Random Number Functions
foreign import ccall unsafe "__gmpz_urandomb" mpz_urandomb :: Ptr MPZ -> Ptr ↙
    ↳ GMPRandState -> MPBitCnt -> IO ()
foreign import ccall unsafe "__gmpz_urandomm" mpz_urandomm :: Ptr MPZ -> Ptr ↙
    ↳ GMPRandState -> SrcPtr MPZ -> IO ()
foreign import ccall unsafe "__gmpz_rrandomb" mpz_rrandomb :: Ptr MPZ -> Ptr ↙
    ↳ GMPRandState -> MPBitCnt -> IO ()
-- ** Integer Import and Export
foreign import ccall unsafe "__gmpz_import" mpz_import :: Ptr MPZ -> CSize -> ↙
    ↳ CInt -> CSize -> CInt -> CSize -> Ptr a -> IO ()
foreign import ccall unsafe "__gmpz_export" mpz_export :: Ptr a -> Ptr CSize -> ↙
    ↳ CInt -> CSize -> CInt -> CSize -> SrcPtr MPZ -> IO ()
-- ** Miscellaneous Functions
-- mpz_fits_ulong_p
foreign import ccall unsafe "__gmpz_fits_slong_p" mpz_fits_slong_p :: SrcPtr MPZ↙
    ↳ -> IO CInt
-- mpz_fits_uint_p
foreign import ccall unsafe "__gmpz_fits_sint_p" mpz_fits_sint_p :: SrcPtr MPZ ↙
    ↳ -> IO CInt
-- mpz_fits_ushort_p
foreign import ccall unsafe "__gmpz_fits_sshort_p" mpz_fits_sshort_p :: SrcPtr ↙
    ↳ MPZ -> IO CInt
-- mpz_odd_p
-- mpz_even_p
foreign import ccall unsafe "__gmpz_sizeinbase" mpz_sizeinbase :: SrcPtr MPZ -> ↙
    ↳ CInt -> IO CSize
-- ** Special Functions
foreign import ccall unsafe "__gmpz_realloc" mpz_realloc :: Ptr MPZ -> MPSize ->↙
    ↳ IO ()
-- mpz_get_limbn
```

```
    -- mpz_size
    foreign import ccall unsafe "__gmpz_limbs_read" mpz_limbs_read :: SrcPtr MPZ -> ↙
        ↳ IO (SrcPtr MPLimb)
185 foreign import ccall unsafe "__gmpz_limbs_write" mpz_limbs_write :: Ptr MPZ -> ↙
        ↳ MPSize -> IO (Ptr MPLimb)
    foreign import ccall unsafe "__gmpz_limbs_modify" mpz_limbs_modify :: Ptr MPZ ->↙
        ↳  MPSize -> IO (Ptr MPLimb)
    foreign import ccall unsafe "__gmpz_limbs_finish" mpz_limbs_finish :: Ptr MPZ ->↙
        ↳  MPSize -> IO ()
    foreign import ccall unsafe "__gmpz_roinit_n" mpz_roinit_n :: Ptr MPZ -> SrcPtr ↙
        ↳ MPLimb -> MPSize -> IO (SrcPtr MPZ)
    -- MPZ_ROINIT_N
190
    -- * Rational Number Functions
    foreign import ccall unsafe "__gmpq_canonicalize" mpq_canonicalize :: Ptr MPQ ->↙
        ↳  IO ()
    -- ** Initialization and Assignment Functions
    foreign import ccall unsafe "__gmpq_init" mpq_init :: Ptr MPQ -> IO ()
195 foreign import ccall unsafe "__gmpq_clear" mpq_clear :: Ptr MPQ -> IO ()
    foreign import ccall unsafe "__gmpq_set" mpq_set :: Ptr MPQ -> SrcPtr MPQ -> IO ↙
        ↳ ()
    foreign import ccall unsafe "__gmpq_set_z" mpq_set_z :: Ptr MPQ -> SrcPtr MPZ ->↙
        ↳  IO ()
    foreign import ccall unsafe "__gmpq_set_ui" mpq_set_ui :: Ptr MPQ -> CULong -> ↙
        ↳ CULong -> IO ()
    foreign import ccall unsafe "__gmpq_set_si" mpq_set_si :: Ptr MPQ -> CLong -> ↙
        ↳ CULong -> IO ()
200 foreign import ccall unsafe "__gmpq_set_str" mpq_set_str :: Ptr MPQ -> SrcPtr ↙
        ↳ CChar -> CInt -> IO CInt
    foreign import ccall unsafe "__gmpq_swap" mpq_swap :: Ptr MPQ -> Ptr MPQ -> IO ↙
        ↳ ()
    -- ** Conversion Functions
    foreign import ccall unsafe "__gmpq_get_d" mpq_get_d :: SrcPtr MPQ -> IO CDouble
    foreign import ccall unsafe "__gmpq_set_d" mpq_set_d :: Ptr MPQ -> CDouble -> IO↙
        ↳  ()
205 foreign import ccall unsafe "__gmpq_set_f" mpq_set_f :: Ptr MPQ -> SrcPtr MPF ->↙
        ↳  IO ()
    foreign import ccall unsafe "__gmpq_get_str" mpq_get_str :: Ptr CChar -> CInt ->↙
        ↳  SrcPtr MPQ -> IO (Ptr CChar)
    -- ** Arithmetic Functions
    foreign import ccall unsafe "__gmpq_add" mpq_add :: Ptr MPQ -> SrcPtr MPQ -> ↙
        ↳ SrcPtr MPQ -> IO ()
    foreign import ccall unsafe "__gmpq_sub" mpq_sub :: Ptr MPQ -> SrcPtr MPQ -> ↙
        ↳ SrcPtr MPQ -> IO ()
210 foreign import ccall unsafe "__gmpq_mul" mpq_mul :: Ptr MPQ -> SrcPtr MPQ -> ↙
        ↳ SrcPtr MPQ -> IO ()
    foreign import ccall unsafe "__gmpq_mul_2exp" mpq_mul_2exp :: Ptr MPQ -> SrcPtr ↙
        ↳ MPQ -> MPBitCnt -> IO ()
    foreign import ccall unsafe "__gmpq_div" mpq_div :: Ptr MPQ -> SrcPtr MPQ -> ↙
        ↳ SrcPtr MPQ -> IO ()
    foreign import ccall unsafe "__gmpq_div_2exp" mpq_div_2exp :: Ptr MPQ -> SrcPtr ↙
        ↳ MPQ -> MPBitCnt -> IO ()
    -- mpq_neg
215 -- mpq_abs
    foreign import ccall unsafe "__gmpq_inv" mpq_inv :: Ptr MPQ -> SrcPtr MPQ -> IO ↙
        ↳ ()
    -- ** Comparison Functions
```

```
foreign import ccall unsafe "__gmpq_cmp" mpq_cmp :: SrcPtr MPQ -> SrcPtr MPQ -> ↙
    ↳ IO CInt
foreign import ccall unsafe "__gmpq_cmp_z" mpq_cmp_z :: SrcPtr MPQ -> SrcPtr MPZ↙
    ↳  -> IO CInt
foreign import ccall unsafe "__gmpq_cmp_ui" mpq_cmp_ui :: SrcPtr MPQ -> CULong ↙
    ↳ -> CULong -> IO CInt
foreign import ccall unsafe "__gmpq_cmp_si" mpq_cmp_si :: SrcPtr MPQ -> CLong ->↙
    ↳  CULong -> IO CInt
-- mpq_sgn
foreign import ccall unsafe "__gmpq_equal" mpq_equal :: SrcPtr MPQ -> SrcPtr MPQ↙
    ↳  -> IO CInt
-- ** Applying Integer Functions to Rationals
-- See also 'mpq_numref' and 'mpq_denref'.
foreign import ccall unsafe "__gmpq_get_num" mpq_get_num :: Ptr MPZ -> SrcPtr ↙
    ↳ MPQ -> IO ()
foreign import ccall unsafe "__gmpq_get_den" mpq_get_den :: Ptr MPZ -> SrcPtr ↙
    ↳ MPQ -> IO ()
foreign import ccall unsafe "__gmpq_set_num" mpq_set_num :: Ptr MPQ -> SrcPtr ↙
    ↳ MPZ -> IO ()
foreign import ccall unsafe "__gmpq_set_den" mpq_set_den :: Ptr MPQ -> SrcPtr ↙
    ↳ MPZ -> IO ()
-- ** Input and Output Functions
foreign import ccall unsafe "__gmpq_out_str" mpq_out_str :: Ptr CFile -> CInt ->↙
    ↳  SrcPtr MPQ -> IO CSize
foreign import ccall unsafe "__gmpq_inp_str" mpq_inp_str :: Ptr MPQ -> Ptr CFile↙
    ↳  -> CInt -> IO CSize

-- * Floating-point Functions
-- ** Initialization Functions
{-
-- not thread-safe, ie, requires running everything in a bound thread for ↙
    ↳ expected behaviour
foreign import ccall unsafe "__gmpf_set_default_prec" mpf_set_default_prec :: ↙
    ↳ MPBitCnt -> IO ()
foreign import ccall unsafe "__gmpf_get_default_prec" mpf_get_default_prec :: IO↙
    ↳  MPBitCnt
foreign import ccall unsafe "__gmpf_init" mpf_init :: Ptr MPF -> IO ()
-}
foreign import ccall unsafe "__gmpf_init2" mpf_init2 :: Ptr MPF -> MPBitCnt -> ↙
    ↳ IO ()
foreign import ccall unsafe "__gmpf_clear" mpf_clear :: Ptr MPF -> IO ()
foreign import ccall unsafe "__gmpf_get_prec" mpf_get_prec :: SrcPtr MPF -> IO ↙
    ↳ MPBitCnt
foreign import ccall unsafe "__gmpf_set_prec" mpf_set_prec :: Ptr MPF -> ↙
    ↳ MPBitCnt -> IO ()
foreign import ccall unsafe "__gmpf_set_prec_raw" mpf_set_prec_raw :: Ptr MPF ->↙
    ↳  MPBitCnt -> IO ()
-- ** Assignment Functions
foreign import ccall unsafe "__gmpf_set" mpf_set :: Ptr MPF -> SrcPtr MPF -> IO ↙
    ↳ ()
foreign import ccall unsafe "__gmpf_set_ui" mpf_set_ui :: Ptr MPF -> CULong -> ↙
    ↳ IO ()
foreign import ccall unsafe "__gmpf_set_si" mpf_set_si :: Ptr MPF -> CLong -> IO↙
    ↳  ()
foreign import ccall unsafe "__gmpf_set_d" mpf_set_d :: Ptr MPF -> CDouble -> IO↙
    ↳  ()
foreign import ccall unsafe "__gmpf_set_z" mpf_set_z :: Ptr MPF -> SrcPtr MPZ ->↙
```

```
    ↳ IO ()
foreign import ccall unsafe "__gmpf_set_q" mpf_set_q :: Ptr MPF -> SrcPtr MPQ ->↙
    ↳ IO ()
foreign import ccall unsafe "__gmpf_set_str" mpf_set_str :: Ptr MPF -> SrcPtr ↙
    ↳ CChar -> CInt -> IO CInt
foreign import ccall unsafe "__gmpf_swap" mpf_swap :: Ptr MPF -> Ptr MPF -> IO ↙
    ↳ ()
-- ** Combined Initialization and Assignment Functions
foreign import ccall unsafe "__gmpf_init_set" mpf_init_set :: Ptr MPF -> SrcPtr ↙
    ↳ MPF -> IO ()
foreign import ccall unsafe "__gmpf_init_set_ui" mpf_init_set_ui :: Ptr MPF -> ↙
    ↳ CULong -> IO ()
foreign import ccall unsafe "__gmpf_init_set_si" mpf_init_set_si :: Ptr MPF -> ↙
    ↳ CLong -> IO ()
foreign import ccall unsafe "__gmpf_init_set_d" mpf_init_set_d :: Ptr MPF -> ↙
    ↳ CDouble -> IO ()
foreign import ccall unsafe "__gmpf_init_set_str" mpf_init_set_str :: Ptr MPF ->↙
    ↳ SrcPtr CChar -> CInt -> IO CInt
-- ** Conversion Functions
foreign import ccall unsafe "__gmpf_get_d" mpf_get_d :: SrcPtr MPF -> IO CDouble
foreign import ccall unsafe "__gmpf_get_d_2exp" mpf_get_d_2exp :: Ptr CLong -> ↙
    ↳ SrcPtr MPF -> IO CDouble
foreign import ccall unsafe "__gmpf_get_si" mpf_get_si :: SrcPtr MPF -> IO CLong
foreign import ccall unsafe "__gmpf_get_ui" mpf_get_ui :: SrcPtr MPF -> IO ↙
    ↳ CULong
foreign import ccall unsafe "__gmpf_get_str" mpf_get_str :: Ptr CChar -> Ptr ↙
    ↳ MPExp -> CInt -> CSize -> SrcPtr MPF -> IO (Ptr CChar)
-- ** Arithmetic Functions
foreign import ccall unsafe "__gmpf_add" mpf_add :: Ptr MPF -> SrcPtr MPF -> ↙
    ↳ SrcPtr MPF -> IO ()
foreign import ccall unsafe "__gmpf_add_ui" mpf_add_ui :: Ptr MPF -> SrcPtr MPF ↙
    ↳ -> CULong -> IO ()
foreign import ccall unsafe "__gmpf_sub" mpf_sub :: Ptr MPF -> SrcPtr MPF -> ↙
    ↳ SrcPtr MPF -> IO ()
foreign import ccall unsafe "__gmpf_sub_ui" mpf_sub_ui :: Ptr MPF -> SrcPtr MPF ↙
    ↳ -> CULong -> IO ()
foreign import ccall unsafe "__gmpf_ui_sub" mpf_ui_sub :: Ptr MPF -> CULong -> ↙
    ↳ SrcPtr MPF -> IO ()
foreign import ccall unsafe "__gmpf_mul" mpf_mul :: Ptr MPF -> SrcPtr MPF -> ↙
    ↳ SrcPtr MPF -> IO ()
foreign import ccall unsafe "__gmpf_mul_ui" mpf_mul_ui :: Ptr MPF -> SrcPtr MPF ↙
    ↳ -> CULong -> IO ()
foreign import ccall unsafe "__gmpf_div" mpf_div :: Ptr MPF -> SrcPtr MPF -> ↙
    ↳ SrcPtr MPF -> IO ()
foreign import ccall unsafe "__gmpf_ui_div" mpf_ui_div :: Ptr MPF -> CULong -> ↙
    ↳ SrcPtr MPF -> IO ()
foreign import ccall unsafe "__gmpf_div_ui" mpf_div_ui :: Ptr MPF -> SrcPtr MPF ↙
    ↳ -> CULong -> IO ()
foreign import ccall unsafe "__gmpf_sqrt" mpf_sqrt :: Ptr MPF -> SrcPtr MPF -> ↙
    ↳ IO ()
foreign import ccall unsafe "__gmpf_sqrt_ui" mpf_sqrt_ui :: Ptr MPF -> CULong ->↙
    ↳ IO ()
foreign import ccall unsafe "__gmpf_pow_ui" mpf_pow_ui :: Ptr MPF -> SrcPtr MPF ↙
    ↳ -> CULong -> IO ()
foreign import ccall unsafe "__gmpf_neg" mpf_neg :: Ptr MPF -> SrcPtr MPF -> IO ↙
    ↳ ()
foreign import ccall unsafe "__gmpf_abs" mpf_abs :: Ptr MPF -> SrcPtr MPF -> IO ↙
```

```
    ↳ ()
foreign import ccall unsafe "__gmpf_mul_2exp" mpf_mul_2exp :: Ptr MPF -> SrcPtr ↙
    ↳ MPF -> MPBitCnt -> IO ()
foreign import ccall unsafe "__gmpf_div_2exp" mpf_div_2exp :: Ptr MPF -> SrcPtr ↙
    ↳ MPF -> MPBitCnt -> IO ()
-- ** Comparison Functions
foreign import ccall unsafe "__gmpf_cmp" mpf_cmp :: SrcPtr MPF -> SrcPtr MPF -> ↙
    ↳ IO CInt
foreign import ccall unsafe "__gmpf_cmp_z" mpf_cmp_z :: SrcPtr MPF -> SrcPtr MPZ↙
    ↳  -> IO CInt
foreign import ccall unsafe "__gmpf_cmp_d" mpf_cmp_d :: SrcPtr MPF -> CDouble ->↙
    ↳  IO CInt
foreign import ccall unsafe "__gmpf_cmp_ui" mpf_cmp_ui :: SrcPtr MPF -> CULong ↙
    ↳ -> IO CInt
foreign import ccall unsafe "__gmpf_cmp_si" mpf_cmp_si :: SrcPtr MPF -> CLong ->↙
    ↳  IO CInt
foreign import ccall unsafe "__gmpf_reldiff" mpf_reldiff :: Ptr MPF -> SrcPtr ↙
    ↳ MPF -> SrcPtr MPF -> IO ()
-- mpf_sgn
-- ** Input and Output Functions
foreign import ccall unsafe "__gmpf_out_str" mpf_out_str :: Ptr CFile -> CInt ->↙
    ↳  CSize -> SrcPtr MPF -> IO CSize
foreign import ccall unsafe "__gmpf_inp_str" mpf_inp_str :: Ptr MPF -> Ptr CFile↙
    ↳  -> CInt -> IO CSize
-- ** Miscellaneous Functions
foreign import ccall unsafe "__gmpf_ceil" mpf_ceil :: Ptr MPF -> SrcPtr MPF -> ↙
    ↳ IO ()
foreign import ccall unsafe "__gmpf_floor" mpf_floor :: Ptr MPF -> SrcPtr MPF ->↙
    ↳  IO ()
foreign import ccall unsafe "__gmpf_trunc" mpf_trunc :: Ptr MPF -> SrcPtr MPF ->↙
    ↳  IO ()
foreign import ccall unsafe "__gmpf_integer_p" mpf_integer_p :: SrcPtr MPF -> IO↙
    ↳  CInt
foreign import ccall unsafe "__gmpf_fits_ulong_p" mpf_fits_ulong_p :: SrcPtr MPF↙
    ↳  -> IO CInt
foreign import ccall unsafe "__gmpf_fits_slong_p" mpf_fits_slong_p :: SrcPtr MPF↙
    ↳  -> IO CInt
foreign import ccall unsafe "__gmpf_fits_uint_p" mpf_fits_uint_p :: SrcPtr MPF ↙
    ↳ -> IO CInt
foreign import ccall unsafe "__gmpf_fits_sint_p" mpf_fits_sint_p :: SrcPtr MPF ↙
    ↳ -> IO CInt
foreign import ccall unsafe "__gmpf_fits_ushort_p" mpf_fits_ushort_p :: SrcPtr ↙
    ↳ MPF -> IO CInt
foreign import ccall unsafe "__gmpf_fits_sshort_p" mpf_fits_sshort_p :: SrcPtr ↙
    ↳ MPF -> IO CInt
foreign import ccall unsafe "__gmpf_urandomb" mpf_urandomb :: Ptr MPF -> Ptr ↙
    ↳ GMPRandState -> MPBitCnt -> IO ()
foreign import ccall unsafe "__gmpf_random2" mpf_random2 :: Ptr MPF -> MPSize ->↙
    ↳  MPExp -> IO ()

-- * Random Number Functions
-- ** Random State Initialization
foreign import ccall unsafe "__gmp_randinit_default" gmp_randinit_default :: Ptr↙
    ↳  GMPRandState -> IO ()
foreign import ccall unsafe "__gmp_randinit_mt" gmp_randinit_mt :: Ptr ↙
    ↳ GMPRandState -> IO ()
foreign import ccall unsafe "__gmp_randinit_lc_2exp" gmp_randinit_lc_2exp :: Ptr↙
```

```
    ↳  GMPRandState -> SrcPtr MPZ -> CULong -> MPBitCnt -> IO ()
foreign import ccall unsafe "__gmp_randinit_lc_2exp_size" ↙
    ↳ gmp_randinit_lc_2exp_size :: Ptr GMPRandState -> MPBitCnt -> IO CInt
foreign import ccall unsafe "__gmp_randinit_set" gmp_randinit_set :: Ptr ↙
    ↳ GMPRandState -> SrcPtr GMPRandState -> IO ()
foreign import ccall unsafe "__gmp_randclear" gmp_randclear :: Ptr GMPRandState ↙
    ↳ -> IO ()
-- ** Random State Seeding
foreign import ccall unsafe "__gmp_randseed" gmp_randseed :: Ptr GMPRandState ->↙
    ↳  SrcPtr MPZ -> IO ()
foreign import ccall unsafe "__gmp_randseed_ui" gmp_randseed_ui :: Ptr ↙
    ↳ GMPRandState -> CULong -> IO ()
-- ** Random State Miscellaneous
foreign import ccall unsafe "__gmp_urandomb_ui" gmp_urandomb_ui :: Ptr ↙
    ↳ GMPRandState -> CULong -> IO CULong
foreign import ccall unsafe "__gmp_urandomm_ui" gmp_urandomm_ui :: Ptr ↙
    ↳ GMPRandState -> CULong -> IO CULong

-- * Low-level Functions
foreign import ccall unsafe "__gmpn_add_n" mpn_add_n :: Ptr MPLimb -> SrcPtr ↙
    ↳ MPLimb -> SrcPtr MPLimb -> MPSize -> IO MPLimb
foreign import ccall unsafe "__gmpn_addmul_1" mpn_addmul_1 :: Ptr MPLimb -> ↙
    ↳ SrcPtr MPLimb -> MPSize -> MPLimb -> IO MPLimb
foreign import ccall unsafe "__gmpn_divexact_1" mpn_divexact_1 :: Ptr MPLimb -> ↙
    ↳ SrcPtr MPLimb -> MPSize -> MPLimb -> IO ()
foreign import ccall unsafe "__gmpn_divexact_by3c" mpn_divexact_by3c :: Ptr ↙
    ↳ MPLimb -> SrcPtr MPLimb -> MPSize -> MPLimb -> IO MPLimb
foreign import ccall unsafe "__gmpn_divrem" mpn_divrem :: Ptr MPLimb -> MPSize ↙
    ↳ -> Ptr MPLimb -> MPSize -> SrcPtr MPLimb -> MPSize -> IO MPLimb
foreign import ccall unsafe "__gmpn_divrem_1" mpn_divrem_1 :: Ptr MPLimb -> ↙
    ↳ MPSize -> SrcPtr MPLimb -> MPSize -> MPLimb -> IO MPLimb
foreign import ccall unsafe "__gmpn_divrem_2" mpn_divrem_2 :: Ptr MPLimb -> ↙
    ↳ MPSize -> Ptr MPLimb -> MPSize -> SrcPtr MPLimb -> IO MPLimb
foreign import ccall unsafe "__gmpn_div_qr_1" mpn_div_qr_1 :: Ptr MPLimb -> Ptr ↙
    ↳ MPLimb -> SrcPtr MPLimb -> MPSize -> MPLimb -> IO MPLimb
foreign import ccall unsafe "__gmpn_div_qr_2" mpn_div_qr_2 :: Ptr MPLimb -> Ptr ↙
    ↳ MPLimb -> SrcPtr MPLimb -> MPSize -> SrcPtr MPLimb -> IO MPLimb
foreign import ccall unsafe "__gmpn_gcd" mpn_gcd :: Ptr MPLimb -> Ptr MPLimb -> ↙
    ↳ MPSize -> Ptr MPLimb -> MPSize -> IO MPSize
foreign import ccall unsafe "__gmpn_gcd_1" mpn_gcd_1 :: SrcPtr MPLimb -> MPSize ↙
    ↳ -> MPLimb -> IO MPLimb
foreign import ccall unsafe "__gmpn_gcdext_1" mpn_gcdext_1 :: Ptr MPLimbSigned ↙
    ↳ -> Ptr MPLimbSigned -> MPLimb -> MPLimb -> IO MPLimb
foreign import ccall unsafe "__gmpn_gcdext" mpn_gcdext :: Ptr MPLimb -> Ptr ↙
    ↳ MPLimb -> Ptr MPSize -> Ptr MPLimb -> MPSize -> Ptr MPLimb -> MPSize -> IO↙
    ↳  MPSize
foreign import ccall unsafe "__gmpn_get_str" mpn_get_str :: Ptr CUChar -> CInt ↙
    ↳ -> Ptr MPLimb -> MPSize -> IO CSize
foreign import ccall unsafe "__gmpn_hamdist" mpn_hamdist :: SrcPtr MPLimb -> ↙
    ↳ SrcPtr MPLimb -> MPSize -> IO MPBitCnt
foreign import ccall unsafe "__gmpn_lshift" mpn_lshift :: Ptr MPLimb -> SrcPtr ↙
    ↳ MPLimb -> MPSize -> CUInt -> IO MPLimb
foreign import ccall unsafe "__gmpn_mod_1" mpn_mod_1 :: SrcPtr MPLimb -> MPSize ↙
    ↳ -> MPLimb -> IO MPLimb
foreign import ccall unsafe "__gmpn_mul" mpn_mul :: Ptr MPLimb -> SrcPtr MPLimb ↙
    ↳ -> MPSize -> SrcPtr MPLimb -> MPSize -> IO MPLimb
foreign import ccall unsafe "__gmpn_mul_1" mpn_mul_1 :: Ptr MPLimb -> SrcPtr ↙
```

```
    ↳ MPLimb -> MPSize -> MPLimb -> IO MPLimb
foreign import ccall unsafe "__gmpn_mul_n" mpn_mul_n :: Ptr MPLimb -> SrcPtr ↙
    ↳ MPLimb -> SrcPtr MPLimb -> MPSize -> IO ()
foreign import ccall unsafe "__gmpn_sqr" mpn_sqr :: Ptr MPLimb -> SrcPtr MPLimb ↙
    ↳ -> MPSize -> IO ()
foreign import ccall unsafe "__gmpn_com" mpn_com :: Ptr MPLimb -> SrcPtr MPLimb ↙
    ↳ -> MPSize -> IO ()
foreign import ccall unsafe "__gmpn_perfect_square_p" mpn_perfect_square_p :: ↙
    ↳ SrcPtr MPLimb -> MPSize -> IO CInt
foreign import ccall unsafe "__gmpn_perfect_power_p" mpn_perfect_power_p :: ↙
    ↳ SrcPtr MPLimb -> MPSize -> IO CInt
foreign import ccall unsafe "__gmpn_popcount" mpn_popcount :: SrcPtr MPLimb -> ↙
    ↳ MPSize -> IO MPBitCnt
foreign import ccall unsafe "__gmpn_pow_1" mpn_pow_1 :: Ptr MPLimb -> SrcPtr ↙
    ↳ MPLimb -> MPSize -> MPLimb -> Ptr MPLimb -> IO MPSize
foreign import ccall unsafe "__gmpn_preinv_mod_1" mpn_preinv_mod_1 :: SrcPtr ↙
    ↳ MPLimb -> MPSize -> MPLimb -> MPLimb -> IO MPLimb
foreign import ccall unsafe "__gmpn_random" mpn_random :: Ptr MPLimb -> MPSize ↙
    ↳ -> IO ()
foreign import ccall unsafe "__gmpn_random2" mpn_random2 :: Ptr MPLimb -> MPSize↙
    ↳  -> IO ()
foreign import ccall unsafe "__gmpn_rshift" mpn_rshift :: Ptr MPLimb -> SrcPtr ↙
    ↳ MPLimb -> MPSize -> CUInt -> IO MPLimb
foreign import ccall unsafe "__gmpn_scan0" mpn_scan0 :: SrcPtr MPLimb -> ↙
    ↳ MPBitCnt -> IO MPBitCnt
foreign import ccall unsafe "__gmpn_scan1" mpn_scan1 :: SrcPtr MPLimb -> ↙
    ↳ MPBitCnt -> IO MPBitCnt
foreign import ccall unsafe "__gmpn_set_str" mpn_set_str :: Ptr MPLimb -> SrcPtr↙
    ↳  CChar -> CSize -> CInt -> IO MPSize
foreign import ccall unsafe "__gmpn_sizeinbase" mpn_sizeinbase :: SrcPtr MPLimb ↙
    ↳ -> MPSize -> CInt -> IO CSize
foreign import ccall unsafe "__gmpn_sqrtrem" mpn_sqrtrem :: Ptr MPLimb -> Ptr ↙
    ↳ MPLimb -> SrcPtr MPLimb -> MPSize -> IO MPSize
foreign import ccall unsafe "__gmpn_sub_n" mpn_sub_n :: Ptr MPLimb -> SrcPtr ↙
    ↳ MPLimb -> SrcPtr MPLimb -> MPSize -> IO MPLimb
foreign import ccall unsafe "__gmpn_submul_1" mpn_submul_1 :: Ptr MPLimb -> ↙
    ↳ SrcPtr MPLimb -> MPSize -> MPLimb -> IO MPLimb
foreign import ccall unsafe "__gmpn_tdiv_qr" mpn_tdiv_qr :: Ptr MPLimb -> Ptr ↙
    ↳ MPLimb -> MPSize -> SrcPtr MPLimb -> MPSize -> SrcPtr MPLimb -> MPSize -> ↙
    ↳ IO ()
foreign import ccall unsafe "__gmpn_and_n" mpn_and_n :: Ptr MPLimb -> SrcPtr ↙
    ↳ MPLimb -> SrcPtr MPLimb -> MPSize -> IO ()
foreign import ccall unsafe "__gmpn_andn_n" mpn_andn_n :: Ptr MPLimb -> SrcPtr ↙
    ↳ MPLimb -> SrcPtr MPLimb -> MPSize -> IO ()
foreign import ccall unsafe "__gmpn_nand_n" mpn_nand_n :: Ptr MPLimb -> SrcPtr ↙
    ↳ MPLimb -> SrcPtr MPLimb -> MPSize -> IO ()
foreign import ccall unsafe "__gmpn_ior_n" mpn_ior_n :: Ptr MPLimb -> SrcPtr ↙
    ↳ MPLimb -> SrcPtr MPLimb -> MPSize -> IO ()
foreign import ccall unsafe "__gmpn_iorn_n" mpn_iorn_n :: Ptr MPLimb -> SrcPtr ↙
    ↳ MPLimb -> SrcPtr MPLimb -> MPSize -> IO ()
foreign import ccall unsafe "__gmpn_nior_n" mpn_nior_n :: Ptr MPLimb -> SrcPtr ↙
    ↳ MPLimb -> SrcPtr MPLimb -> MPSize -> IO ()
foreign import ccall unsafe "__gmpn_xor_n" mpn_xor_n :: Ptr MPLimb -> SrcPtr ↙
    ↳ MPLimb -> SrcPtr MPLimb -> MPSize -> IO ()
foreign import ccall unsafe "__gmpn_xnor_n" mpn_xnor_n :: Ptr MPLimb -> SrcPtr ↙
    ↳ MPLimb -> SrcPtr MPLimb -> MPSize -> IO ()
foreign import ccall unsafe "__gmpn_copyi" mpn_copyi :: Ptr MPLimb -> SrcPtr ↙
```

```
    ↳ MPLimb -> MPSize -> IO ()
foreign import ccall unsafe "__gmpn_copyd" mpn_copyd :: Ptr MPLimb -> SrcPtr ↙
    ↳ MPLimb -> MPSize -> IO ()
foreign import ccall unsafe "__gmpn_zero" mpn_zero :: Ptr MPLimb -> MPSize -> IO↙
    ↳  ()
foreign import ccall unsafe "__gmpn_cnd_add_n" mpn_cnd_add_n :: MPLimb -> Ptr ↙
    ↳ MPLimb -> SrcPtr MPLimb -> SrcPtr MPLimb -> MPSize -> IO MPLimb
foreign import ccall unsafe "__gmpn_cnd_sub_n" mpn_cnd_sub_n :: MPLimb -> Ptr ↙
    ↳ MPLimb -> SrcPtr MPLimb -> SrcPtr MPLimb -> MPSize -> IO MPLimb
foreign import ccall unsafe "__gmpn_sec_add_1" mpn_sec_add_1 :: Ptr MPLimb -> ↙
    ↳ SrcPtr MPLimb -> MPSize -> MPLimb -> Ptr MPLimb -> IO MPLimb
foreign import ccall unsafe "__gmpn_sec_add_1_itch" mpn_sec_add_1_itch :: MPSize↙
    ↳  -> IO MPSize
foreign import ccall unsafe "__gmpn_sec_sub_1" mpn_sec_sub_1 :: Ptr MPLimb -> ↙
    ↳ SrcPtr MPLimb -> MPSize -> MPLimb -> Ptr MPLimb -> IO MPLimb
foreign import ccall unsafe "__gmpn_sec_sub_1_itch" mpn_sec_sub_1_itch :: MPSize↙
    ↳  -> IO MPSize
foreign import ccall unsafe "__gmpn_cnd_swap" mpn_cnd_swap :: MPLimb -> ↙
    ↳ VolatilePtr MPLimb -> VolatilePtr MPLimb -> MPSize -> IO ()
foreign import ccall unsafe "__gmpn_sec_mul" mpn_sec_mul :: Ptr MPLimb -> SrcPtr↙
    ↳  MPLimb -> MPSize -> SrcPtr MPLimb -> MPSize -> Ptr MPLimb -> IO ()
foreign import ccall unsafe "__gmpn_sec_mul_itch" mpn_sec_mul_itch :: MPSize -> ↙
    ↳ MPSize -> IO MPSize
foreign import ccall unsafe "__gmpn_sec_sqr" mpn_sec_sqr :: Ptr MPLimb -> SrcPtr↙
    ↳  MPLimb -> MPSize -> Ptr MPLimb -> IO ()
foreign import ccall unsafe "__gmpn_sec_sqr_itch" mpn_sec_sqr_itch :: MPSize -> ↙
    ↳ IO MPSize
foreign import ccall unsafe "__gmpn_sec_powm" mpn_sec_powm :: Ptr MPLimb -> ↙
    ↳ SrcPtr MPLimb -> MPSize -> SrcPtr MPLimb -> MPBitCnt -> SrcPtr MPLimb -> ↙
    ↳ MPSize -> Ptr MPLimb -> IO ()
foreign import ccall unsafe "__gmpn_sec_powm_itch" mpn_sec_powm_itch :: MPSize ↙
    ↳ -> MPBitCnt -> MPSize -> IO MPSize
foreign import ccall unsafe "__gmpn_sec_tabselect" mpn_sec_tabselect :: ↙
    ↳ VolatilePtr MPLimb -> VolatileSrcPtr MPLimb -> MPSize -> MPSize -> MPSize ↙
    ↳ -> IO ()
foreign import ccall unsafe "__gmpn_sec_div_qr" mpn_sec_div_qr :: Ptr MPLimb -> ↙
    ↳ Ptr MPLimb -> MPSize -> SrcPtr MPLimb -> MPSize -> Ptr MPLimb -> IO MPLimb
foreign import ccall unsafe "__gmpn_sec_div_qr_itch" mpn_sec_div_qr_itch :: ↙
    ↳ MPSize -> MPSize -> IO MPSize
foreign import ccall unsafe "__gmpn_sec_div_r" mpn_sec_div_r :: Ptr MPLimb -> ↙
    ↳ MPSize -> SrcPtr MPLimb -> MPSize -> Ptr MPLimb -> IO ()
foreign import ccall unsafe "__gmpn_sec_div_r_itch" mpn_sec_div_r_itch :: MPSize↙
    ↳  -> MPSize -> IO MPSize
foreign import ccall unsafe "__gmpn_sec_invert" mpn_sec_invert :: Ptr MPLimb -> ↙
    ↳ Ptr MPLimb -> SrcPtr MPLimb -> MPSize -> MPBitCnt -> Ptr MPLimb -> IO CInt
foreign import ccall unsafe "__gmpn_sec_invert_itch" mpn_sec_invert_itch :: ↙
    ↳ MPSize -> IO MPSize

{-
foreign import ccall unsafe "__gmpz_dump" mpz_dump :: SrcPtr MPZ -> IO ()
foreign import ccall unsafe "__gmpz_millerrabin" mpz_millerrabin :: SrcPtr MPZ ↙
    ↳ -> CInt -> IO CInt
foreign import ccall unsafe "__gmpf_dump" mpf_dump :: SrcPtr MPF -> IO ()
foreign import ccall unsafe "__gmpf_eq" mpf_eq :: SrcPtr MPF -> SrcPtr MPF -> ↙
    ↳ MPBitCnt -> IO CInt
foreign import ccall unsafe "__gmpf_size" mpf_size :: SrcPtr MPF -> IO CSize
-}
```

11 src/Numeric/GMP/Types.hsc

```
#include <ghc-gmp.h>

#if __GLASGOW_HASKELL__ < 800
#let alignment t = "%lu", (unsigned long)offsetof(struct {char x__; t (y__); }, ↙
    ↳ y__)
#endif

{-# LANGUAGE DeriveDataTypeable #-}
{-# LANGUAGE GeneralizedNewtypeDeriving #-}

-- | GMP types.
module Numeric.GMP.Types where

import Data.Data
import Data.Typeable
import Data.Bits
import Data.Ix
import Data.Int
import Data.Word

import Foreign (Storable(..), Ptr, nullPtr, plusPtr)
import Foreign.C (CInt)

-- | @mpz_t@
data MPZ = MPZ
  { mpzAlloc :: !CInt
  , mpzSize :: !CInt
  , mpzD :: !(Ptr MPLimb)
  }

instance Storable MPZ where
  sizeOf _ = (#size __mpz_struct)
  alignment _ = (#alignment __mpz_struct)
  peek ptr = do
    alloc <- (#peek __mpz_struct, _mp_alloc) ptr
    size <- (#peek __mpz_struct, _mp_size) ptr
    d <- (#peek __mpz_struct, _mp_d) ptr
    return (MPZ{ mpzAlloc = alloc, mpzSize = size, mpzD = d })
  poke ptr (MPZ{ mpzAlloc = alloc, mpzSize = size, mpzD = d }) = do
    (#poke __mpz_struct, _mp_alloc) ptr alloc
    (#poke __mpz_struct, _mp_size) ptr size
    (#poke __mpz_struct, _mp_d) ptr d

-- | @mpq_t@
data MPQ = MPQ
  { mpqNum :: !MPZ
  , mpqDen :: !MPZ
  }

instance Storable MPQ where
  sizeOf _ = (#size __mpq_struct)
  alignment _ = (#alignment __mpq_struct)
  peek ptr = do
    num <- (#peek __mpq_struct, _mp_num) ptr
    den <- (#peek __mpq_struct, _mp_den) ptr
```

```
55        return (MPQ{ mpqNum = num, mpqDen = den })
      poke ptr (MPQ{ mpqNum = num, mpqDen = den }) = do
        (#poke __mpq_struct, _mp_num) ptr num
        (#poke __mpq_struct, _mp_den) ptr den

60    -- | Get pointers to numerator and denominator (these are macros in the C API).
    mpq_numref, mpq_denref :: Ptr MPQ -> Ptr MPZ
    mpq_numref ptr = plusPtr ptr (#offset __mpq_struct, _mp_num)
    mpq_denref ptr = plusPtr ptr (#offset __mpq_struct, _mp_den)

65  -- | @mpf_t@
    data MPF = MPF
      { mpfPrec :: !CInt
      , mpfSize :: !CInt
      , mpfExp :: !MPExp
70    , mpfD :: !(Ptr MPLimb)
      }

    instance Storable MPF where
      sizeOf _ = (#size __mpf_struct)
75    alignment _ = (#alignment __mpf_struct)
      peek ptr = do
        prec <- (#peek __mpf_struct, _mp_prec) ptr
        size <- (#peek __mpf_struct, _mp_size) ptr
        expo <- (#peek __mpf_struct, _mp_exp) ptr
80      d <- (#peek __mpf_struct, _mp_d) ptr
        return (MPF{ mpfPrec = prec, mpfSize = size, mpfExp = expo, mpfD = d })
      poke ptr (MPF{ mpfPrec = prec, mpfSize = size, mpfExp = expo, mpfD = d }) = do
        (#poke __mpf_struct, _mp_prec) ptr prec
        (#poke __mpf_struct, _mp_size) ptr size
85      (#poke __mpf_struct, _mp_exp) ptr expo
        (#poke __mpf_struct, _mp_d) ptr d

    -- | @gmp_randstate_t@
    data GMPRandState = GMPRandState
90    { gmprsSeed :: !MPZ
      , gmprsAlg :: !GMPRandAlg
      , gmprsAlgData :: !(Ptr ())
      }

95  instance Storable GMPRandState where
      sizeOf _ = (#size __gmp_randstate_struct)
      alignment _ = (#alignment __gmp_randstate_struct)
      peek ptr = do
        seed <- (#peek __gmp_randstate_struct, _mp_seed) ptr
100     alg <- (#peek __gmp_randstate_struct, _mp_alg) ptr
        algdata <- (#peek __gmp_randstate_struct, _mp_algdata._mp_lc) ptr
        return (GMPRandState{ gmprsSeed = seed, gmprsAlg = alg, gmprsAlgData = ↙
            ↳ algdata })
      poke ptr (GMPRandState{ gmprsSeed = seed, gmprsAlg = alg, gmprsAlgData = ↙
          ↳ algdata }) = do
        (#poke __gmp_randstate_struct, _mp_seed) ptr seed
105     (#poke __gmp_randstate_struct, _mp_alg) ptr alg
        (#poke __gmp_randstate_struct, _mp_algdata) ptr algdata

    -- | @mp_limb_t@
    newtype MPLimb = MPLimb (#type mp_limb_t)
```

```
110   deriving (Eq, Ord, Read, Show, Enum, Bounded, Num, Integral, Real, Ix, Bits, ↙
          ↳ FiniteBits, Data, Typeable, Storable)

-- | @mp_limb_signed_t@
newtype MPLimbSigned = MPLimbSigned (#type mp_limb_signed_t)
  deriving (Eq, Ord, Read, Show, Enum, Bounded, Num, Integral, Real, Ix, Bits, ↙
      ↳ FiniteBits, Data, Typeable, Storable)
115
-- | @mp_size_t@
newtype MPSize = MPSize (#type mp_size_t)
  deriving (Eq, Ord, Read, Show, Enum, Bounded, Num, Integral, Real, Ix, Bits, ↙
      ↳ FiniteBits, Data, Typeable, Storable)

120 -- | @mp_exp_t@
newtype MPExp = MPExp (#type mp_exp_t)
  deriving (Eq, Ord, Read, Show, Enum, Bounded, Num, Integral, Real, Ix, Bits, ↙
      ↳ FiniteBits, Data, Typeable, Storable)

-- | @mp_bitcnt_t@
125 newtype MPBitCnt = MPBitCnt (#type mp_bitcnt_t)
  deriving (Eq, Ord, Read, Show, Enum, Bounded, Num, Integral, Real, Ix, Bits, ↙
      ↳ FiniteBits, Data, Typeable, Storable)

-- | @gmp_randalg_t@
newtype GMPRandAlg = GMPRandAlg (#type gmp_randalg_t)
130   deriving (Eq, Ord, Read, Show, Enum, Bounded, Num, Integral, Real, Ix, Bits, ↙
          ↳ FiniteBits, Data, Typeable, Storable)
```

12 src/Numeric/GMP/Utils.hs

```
{-# LANGUAGE ForeignFunctionInterface #-}
{-# LANGUAGE MagicHash #-}
{-# LANGUAGE UnboxedTuples #-}
-- | GMP utilities. A simple example with probable primes:
5 --
-- > import Numeric.GMP.Raw.Safe (mpz_nextprime)
-- >
-- > nextPrime :: Integer -> Integer
-- > nextPrime n =
10 -- >   unsafePerformIO $
-- >     withOutInteger_ $ \rop ->
-- >       withInInteger n $ \op ->
-- >         mpz_nextprime rop op
module Numeric.GMP.Utils
15   ( -- * Integer marshalling
     withInInteger'
   , withInInteger
   , withInOutInteger
   , withInOutInteger_
20   , withOutInteger
   , withOutInteger_
   , peekInteger'
   , peekInteger
   , pokeInteger
25     -- * Rational marshalling
   , withInRational'
   , withInRational
```

```
    , withInOutRational
    , withInOutRational_
    , withOutRational
    , withOutRational_
    , peekRational'
    , peekRational
    , pokeRational
    ) where

import Control.Exception (bracket_)
import Data.Ratio ((%), numerator, denominator)
import Foreign (allocaBytes, alloca, with, sizeOf, peek)

import GHC.Integer.GMP.Internals
  ( Integer(..)
  , BigNat(..)
  , sizeofBigNat#
  , byteArrayToBigNat#
  , bigNatToInteger
  , bigNatToNegInteger
  )
import GHC.Prim
  ( ByteArray#
  , sizeofByteArray#
  , copyByteArrayToAddr#
  , newByteArray#
  , copyAddrToByteArray#
  , unsafeFreezeByteArray#
  )
import GHC.Exts (Int(..), Ptr(..))
import GHC.Types (IO(..))

import Numeric.GMP.Types

import Numeric.GMP.Raw.Unsafe
  ( mpz_init
  , mpz_clear
  , mpq_init
  , mpq_clear
  , mpz_set
  )

foreign import ccall unsafe "mpz_set_HsInt" -- implemented in wrappers.c
  mpz_set_HsInt :: Ptr MPZ -> Int -> IO ()


-- | Store an 'Integer' into a temporary 'MPZ'.  The action must use it only
--   as an @mpz_srcptr@ (ie, constant/immutable), and must not allow references
--   to it to escape its scope.
withInInteger' :: Integer -> (MPZ -> IO r) -> IO r
withInInteger' i action = case i of
  S# n# -> alloca $ \src -> bracket_ (mpz_init src) (mpz_clear src) $ do
    -- a bit awkward, TODO figure out how to do this without foreign calls?
    mpz_set_HsInt src (I# n#)
    z <- peek src
    r <- action z
    return r
```

```
    Jp# bn@(BN# ba#) -> withByteArray ba# $ \d _ -> action MPZ
          { mpzAlloc = 0
          , mpzSize = fromIntegral (I# (sizeofBigNat# bn))
          , mpzD = d
          }
    Jn# bn@(BN# ba#) -> withByteArray ba# $ \d _ -> action MPZ
          { mpzAlloc = 0
          , mpzSize = - fromIntegral (I# (sizeofBigNat# bn))
          , mpzD = d
          }

withByteArray :: ByteArray# -> (Ptr a -> Int -> IO r) -> IO r
withByteArray ba# f = do
  let bytes = I# (sizeofByteArray# ba#)
  allocaBytes bytes $ \ptr@(Ptr addr#) -> do
    IO (\s -> (# copyByteArrayToAddr# ba# 0# addr# (sizeofByteArray# ba#) s, () ↙
        ↳ #))
    f ptr bytes


-- | Combination of 'withInInteger'' and 'with'.  The action must use it only
--   as an @mpz_srcptr@ (ie, constant/immutable), and must not allow the pointer
--   to escape its scope.  If in doubt about potential mutation by the action,
--   use 'withInOutInteger' instead.
withInInteger :: Integer -> (Ptr MPZ -> IO r) -> IO r
withInInteger i action = withInInteger' i $ \z -> with z action


-- | Allocates and initializes an @mpz_t@, pokes the value, and peeks and clears
--   it after the action.  The pointer must not escape the scope of the action.
withInOutInteger :: Integer -> (Ptr MPZ -> IO a) -> IO (Integer, a)
withInOutInteger n action = withOutInteger $ \z -> do
  pokeInteger z n
  action z


-- | Allocates and initializes an @mpz_t@, pokes the value, and peeks and clears
--   it after the action.  The pointer must not escape the scope of the action.
--   The result of the action is discarded.
withInOutInteger_ :: Integer -> (Ptr MPZ -> IO a) -> IO Integer
withInOutInteger_ n action = do
  (z, _) <- withInOutInteger n action
  return z


-- | Allocates and initializes an @mpz_t@, then peeks and clears it after the
--   action.  The pointer must not escape the scope of the action.
withOutInteger :: (Ptr MPZ -> IO a) -> IO (Integer, a)
withOutInteger action = alloca $ \ptr ->
  bracket_ (mpz_init ptr) (mpz_clear ptr) $ do
    a <- action ptr
    z <- peekInteger ptr
    return (z, a)


-- | Allocates and initializes an @mpz_t@, then peeks and clears it after the
--   action.  The pointer must not escape the scope of the action.  The result
```

```
--   of the action is discarded.
withOutInteger_ :: (Ptr MPZ -> IO a) -> IO Integer
withOutInteger_ action = do
  (z, _) <- withOutInteger action
  return z


-- | Store an 'Integer' into an @mpz_t@, which must have been initialized with
--   @mpz_init@.
pokeInteger :: Ptr MPZ -> Integer -> IO ()
pokeInteger dst (S# n#) = mpz_set_HsInt dst (I# n#)
-- copies twice, once in withInteger, and again in @mpz_set@.
-- could maybe rewrite to do one copy, using gmp's own alloc functions?
pokeInteger dst j = withInInteger j $ mpz_set dst


-- | Read an 'Integer' from an 'MPZ'.
peekInteger' :: MPZ -> IO Integer
peekInteger' MPZ{ mpzSize = size, mpzD = d } = do
  if size == 0 then return 0 else
-- This copies once, from 'Ptr' 'MPLimb' to 'ByteArray#'
-- 'byteArrayToBigNat#' hopefully won't need to copy it again
    asByteArray d (fromIntegral (abs size) * sizeOf (undefined :: MPLimb))
      (\ba# -> return $ case fromIntegral (abs size) of
        I# size# -> (if size < 0 then bigNatToNegInteger else bigNatToInteger)
            (byteArrayToBigNat# ba# size#)
      )

asByteArray :: Ptr a -> Int -> (ByteArray# -> IO r) -> IO r
asByteArray (Ptr addr#) (I# bytes#) f = do
  IO $ \s# -> case newByteArray# bytes# s# of
    (# s'#, mba# #) ->
      case unsafeFreezeByteArray# mba# (copyAddrToByteArray# addr# mba# 0# bytes#
          ↪ # s'#) of
        (# s''#, ba# #) -> case f ba# of IO r -> r s''#


-- | Combination of 'peek' and 'peekInteger''.
peekInteger :: Ptr MPZ -> IO Integer
peekInteger src = do
  z <- peek src
  peekInteger' z


-- | Store a 'Rational' into a temporary 'MPQ'.  The action must use it only
--   as an @mpq_srcptr@ (ie, constant/immutable), and must not allow the pointer
--   to escape its scope.
withInRational' :: Rational -> (MPQ -> IO r) -> IO r
withInRational' q action =
  withInInteger' (numerator q) $ \nz ->
  withInInteger' (denominator q) $ \dz ->
  action (MPQ nz dz)


-- | Combination of 'withInRational'' and 'with'.  The action must use it only
--   as an @mpq_srcptr@ (ie, constant/immutable), and must not allow the pointer
--   to escape its scope.  If in doubt about potential mutation by the action,
```

```
--   use 'withInOutRational' instead.
withInRational :: Rational -> (Ptr MPQ -> IO r) -> IO r
withInRational q action = withInRational' q $ \qq -> with qq action

-- | Allocates and initializes an @mpq_t@, pokes the value, and peeks and clears
--   it after the action.  The pointer must not escaep the scope of the action.
withInOutRational :: Rational -> (Ptr MPQ -> IO a) -> IO (Rational, a)
withInOutRational n action = withOutRational $ \q -> do
  pokeRational q n
  action q

-- | Allocates and initializes an @mpq_t@, pokes the value, and peeks and clears
--   it after the action.  The pointer must not escaep the scope of the action.
--   The result of the action is discarded.
withInOutRational_ :: Rational -> (Ptr MPQ -> IO a) -> IO Rational
withInOutRational_ n action = do
  (q, _) <- withInOutRational n action
  return q

-- | Allocates and initializes an @mpq_t@, then peeks and clears it after the
--   action.  The pointer must not escape the scope of the action.
withOutRational :: (Ptr MPQ -> IO a) -> IO (Rational, a)
withOutRational action = alloca $ \ptr ->
  bracket_ (mpq_init ptr) (mpq_clear ptr) $ do
    a <- action ptr
    q <- peekRational ptr
    return (q, a)

-- | Allocates and initializes an @mpq_t@, then peeks and clears it after the
--   action.  The pointer must not escape the scope of the action.  The result
--   of the action is discarded.
withOutRational_ :: (Ptr MPQ -> IO a) -> IO Rational
withOutRational_ action = do
  (q, _) <- withOutRational action
  return q

-- | Store a 'Rational' into an @mpq_t@, which must have been initialized with
--   @mpq_init@.
pokeRational :: Ptr MPQ -> Rational -> IO ()
pokeRational ptr q = do
  pokeInteger (mpq_numref ptr) (numerator q)
  pokeInteger (mpq_denref ptr) (denominator q)

-- | Read a 'Rational' from an 'MPQ'.
peekRational' :: MPQ -> IO Rational
peekRational' (MPQ n d) = do
  num <- peekInteger' n
  den <- peekInteger' d
  return (num % den)
```

```
-- | Combination of 'peek' and 'peekRational''.
peekRational :: Ptr MPQ -> IO Rational
peekRational src = do
  q <- peek src
  peekRational' q
```

13 tests/Main.hs

```
{-# LANGUAGE ForeignFunctionInterface #-}
{-# LANGUAGE TemplateHaskell #-}
import Test.QuickCheck
import Test.QuickCheck.Arbitrary
import Control.Monad (unless)
import System.Exit (exitFailure)
import Foreign
import Numeric.GMP.Types
import Numeric.GMP.Utils
import Numeric.GMP.Raw.Safe (mpz_mul, mpq_mul)


-- instance Arbitrary Integer has small range
newtype Big = Big{ getBig :: Integer } deriving (Show)
instance Arbitrary Big where
  arbitrary = fmap Big $ choose (-bit 100, bit 100)
  shrink = fmap Big . shrinkIntegral . getBig


prop_IntegerWithPeek' n = ioProperty $ do
  m <- withInInteger' n peekInteger'
  return (n == m)

prop_IntegerWithPeek n = ioProperty $ do
  m <- withInInteger n peekInteger
  return (n == m)

prop_IntegerMultiply a b = ioProperty $ do
  (c, _) <-
    withOutInteger $ \cz ->
      withInInteger a $ \az ->
        withInInteger b $ \bz ->
          mpz_mul cz az bz
  return (a * b == c)


prop_BigIntegerWithPeek' (Big n) = ioProperty $ do
  m <- withInInteger' n peekInteger'
  return (n == m)

prop_BigIntegerWithPeek (Big n) = ioProperty $ do
  m <- withInInteger n peekInteger
  return (n == m)

prop_BigIntegerMultiply (Big a) (Big b) = ioProperty $ do
  (c, _) <-
    withOutInteger $ \cz ->
      withInInteger a $ \az ->
        withInInteger b $ \bz ->
```

```
50            mpz_mul cz az bz
  return (a * b == c)


prop_RationalWithPeek' n = ioProperty $ do
55  m <- withInRational' n peekRational'
  return (n == m)

prop_RationalWithPeek n = ioProperty $ do
  m <- withInRational n peekRational
60  return (n == m)

prop_RationalMultiply a b = ioProperty $ do
  (c, _) <-
    withOutRational $ \cq ->
65    withInRational a $ \aq ->
        withInRational b $ \bq ->
          mpq_mul cq aq bq
  return (a * b == c)

70

return []
main = do
  r <- $quickCheckAll
  unless r exitFailure
```