# 1 "/home/drsatya/Desktop/lab1/matrixmul.cpp"

# 1 "<built-in>"

# 1 "<command-line>"

# 1 "/home/drsatya/Desktop/lab1/matrixmul.cpp"

# 1 "/home/drsatya/Desktop/lab1/matrixmul.h" 1

# 69 "/home/drsatya/Desktop/lab1/matrixmul.h"

# 1 "/opt/Xilinx/Vivado\_HLS/2017.2/lnx64/tools/gcc/bin/../lib/gcc/x86\_64-unknown-linux-gnu/4.6.3/../../../../include/c++/4.6.3/cmath" 1 3

# 41 "/opt/Xilinx/Vivado\_HLS/2017.2/lnx64/tools/gcc/bin/../lib/gcc/x86\_64-unknown-linux-gnu/4.6.3/../../../../include/c++/4.6.3/cmath" 3

# 42 "/opt/Xilinx/Vivado\_HLS/2017.2/lnx64/tools/gcc/bin/../lib/gcc/x86\_64-unknown-linux-gnu/4.6.3/../../../../include/c++/4.6.3/cmath" 3

# 1 "/opt/Xilinx/Vivado\_HLS/2017.2/lnx64/tools/gcc/bin/../lib/gcc/x86\_64-unknown-linux-gnu/4.6.3/../../../../include/c++/4.6.3/x86\_64-unknown-linux-gnu/bits/c++config.h" 1 3

# 153 "/opt/Xilinx/Vivado\_HLS/2017.2/lnx64/tools/gcc/bin/../lib/gcc/x86\_64-unknown-linux-gnu/4.6.3/../../../../include/c++/4.6.3/x86\_64-unknown-linux-gnu/bits/c++config.h" 3

namespace std

{

typedef long unsigned int size\_t;

typedef long int ptrdiff\_t;

}

# 393 "/opt/Xilinx/Vivado\_HLS/2017.2/lnx64/tools/gcc/bin/../lib/gcc/x86\_64-unknown-linux-gnu/4.6.3/../../../../include/c++/4.6.3/x86\_64-unknown-linux-gnu/bits/c++config.h" 3

# 1 "/opt/Xilinx/Vivado\_HLS/2017.2/lnx64/tools/gcc/bin/../lib/gcc/x86\_64-unknown-linux-gnu/4.6.3/../../../../include/c++/4.6.3/x86\_64-unknown-linux-gnu/bits/os\_defines.h" 1 3

# 40 "/opt/Xilinx/Vivado\_HLS/2017.2/lnx64/tools/gcc/bin/../lib/gcc/x86\_64-unknown-linux-gnu/4.6.3/../../../../include/c++/4.6.3/x86\_64-unknown-linux-gnu/bits/os\_defines.h" 3

# 1 "/opt/Xilinx/Vivado\_HLS/2017.2/lnx64/tools/gcc/bin/../lib/gcc/x86\_64-unknown-linux-gnu/4.6.3/include-fixed/features.h" 1 3 4

# 339 "/opt/Xilinx/Vivado\_HLS/2017.2/lnx64/tools/gcc/bin/../lib/gcc/x86\_64-unknown-linux-gnu/4.6.3/include-fixed/features.h" 3 4

# 1 "/usr/include/x86\_64-linux-gnu/sys/cdefs.h" 1 3 4

# 378 "/usr/include/x86\_64-linux-gnu/sys/cdefs.h" 3 4

# 1 "/usr/include/x86\_64-linux-gnu/bits/wordsize.h" 1 3 4

# 379 "/usr/include/x86\_64-linux-gnu/sys/cdefs.h" 2 3 4

# 340 "/opt/Xilinx/Vivado\_HLS/2017.2/lnx64/tools/gcc/bin/../lib/gcc/x86\_64-unknown-linux-gnu/4.6.3/include-fixed/features.h" 2 3 4

# 362 "/opt/Xilinx/Vivado\_HLS/2017.2/lnx64/tools/gcc/bin/../lib/gcc/x86\_64-unknown-linux-gnu/4.6.3/include-fixed/features.h" 3 4

# 1 "/usr/include/x86\_64-linux-gnu/gnu/stubs.h" 1 3 4

# 1 "/usr/include/x86\_64-linux-gnu/bits/wordsize.h" 1 3 4

# 5 "/usr/include/x86\_64-linux-gnu/gnu/stubs.h" 2 3 4

# 1 "/usr/include/x86\_64-linux-gnu/gnu/stubs-64.h" 1 3 4

# 10 "/usr/include/x86\_64-linux-gnu/gnu/stubs.h" 2 3 4

# 363 "/opt/Xilinx/Vivado\_HLS/2017.2/lnx64/tools/gcc/bin/../lib/gcc/x86\_64-unknown-linux-gnu/4.6.3/include-fixed/features.h" 2 3 4

# 41 "/opt/Xilinx/Vivado\_HLS/2017.2/lnx64/tools/gcc/bin/../lib/gcc/x86\_64-unknown-linux-gnu/4.6.3/../../../../include/c++/4.6.3/x86\_64-unknown-linux-gnu/bits/os\_defines.h" 2 3

# 394 "/opt/Xilinx/Vivado\_HLS/2017.2/lnx64/tools/gcc/bin/../lib/gcc/x86\_64-unknown-linux-gnu/4.6.3/../../../../include/c++/4.6.3/x86\_64-unknown-linux-gnu/bits/c++config.h" 2 3

# 1 "/opt/Xilinx/Vivado\_HLS/2017.2/lnx64/tools/gcc/bin/../lib/gcc/x86\_64-unknown-linux-gnu/4.6.3/../../../../include/c++/4.6.3/x86\_64-unknown-linux-gnu/bits/cpu\_defines.h" 1 3

# 397 "/opt/Xilinx/Vivado\_HLS/2017.2/lnx64/tools/gcc/bin/../lib/gcc/x86\_64-unknown-linux-gnu/4.6.3/../../../../include/c++/4.6.3/x86\_64-unknown-linux-gnu/bits/c++config.h" 2 3

# 44 "/opt/Xilinx/Vivado\_HLS/2017.2/lnx64/tools/gcc/bin/../lib/gcc/x86\_64-unknown-linux-gnu/4.6.3/../../../../include/c++/4.6.3/cmath" 2 3

# 1 "/opt/Xilinx/Vivado\_HLS/2017.2/lnx64/tools/gcc/bin/../lib/gcc/x86\_64-unknown-linux-gnu/4.6.3/../../../../include/c++/4.6.3/bits/cpp\_type\_traits.h" 1 3

# 36 "/opt/Xilinx/Vivado\_HLS/2017.2/lnx64/tools/gcc/bin/../lib/gcc/x86\_64-unknown-linux-gnu/4.6.3/../../../../include/c++/4.6.3/bits/cpp\_type\_traits.h" 3

# 37 "/opt/Xilinx/Vivado\_HLS/2017.2/lnx64/tools/gcc/bin/../lib/gcc/x86\_64-unknown-linux-gnu/4.6.3/../../../../include/c++/4.6.3/bits/cpp\_type\_traits.h" 3

# 69 "/opt/Xilinx/Vivado\_HLS/2017.2/lnx64/tools/gcc/bin/../lib/gcc/x86\_64-unknown-linux-gnu/4.6.3/../../../../include/c++/4.6.3/bits/cpp\_type\_traits.h" 3

namespace \_\_gnu\_cxx \_\_attribute\_\_ ((\_\_visibility\_\_ ("default")))

{

template<typename \_Iterator, typename \_Container>

class \_\_normal\_iterator;

}

namespace std \_\_attribute\_\_ ((\_\_visibility\_\_ ("default")))

{

struct \_\_true\_type { };

struct \_\_false\_type { };

template<bool>

struct \_\_truth\_type

{ typedef \_\_false\_type \_\_type; };

template<>

struct \_\_truth\_type<true>

{ typedef \_\_true\_type \_\_type; };

template<class \_Sp, class \_Tp>

struct \_\_traitor

{

enum { \_\_value = bool(\_Sp::\_\_value) || bool(\_Tp::\_\_value) };

typedef typename \_\_truth\_type<\_\_value>::\_\_type \_\_type;

};

template<typename, typename>

struct \_\_are\_same

{

enum { \_\_value = 0 };

typedef \_\_false\_type \_\_type;

};

template<typename \_Tp>

struct \_\_are\_same<\_Tp, \_Tp>

{

enum { \_\_value = 1 };

typedef \_\_true\_type \_\_type;

};

template<typename \_Tp>

struct \_\_is\_void

{

enum { \_\_value = 0 };

typedef \_\_false\_type \_\_type;

};

template<>

struct \_\_is\_void<void>

{

enum { \_\_value = 1 };

typedef \_\_true\_type \_\_type;

};

template<typename \_Tp>

struct \_\_is\_integer

{

enum { \_\_value = 0 };

typedef \_\_false\_type \_\_type;

};

template<>

struct \_\_is\_integer<bool>

{

enum { \_\_value = 1 };

typedef \_\_true\_type \_\_type;

};

template<>

struct \_\_is\_integer<char>

{

enum { \_\_value = 1 };

typedef \_\_true\_type \_\_type;

};

template<>

struct \_\_is\_integer<signed char>

{

enum { \_\_value = 1 };

typedef \_\_true\_type \_\_type;

};

template<>

struct \_\_is\_integer<unsigned char>

{

enum { \_\_value = 1 };

typedef \_\_true\_type \_\_type;

};

template<>

struct \_\_is\_integer<wchar\_t>

{

enum { \_\_value = 1 };

typedef \_\_true\_type \_\_type;

};

# 199 "/opt/Xilinx/Vivado\_HLS/2017.2/lnx64/tools/gcc/bin/../lib/gcc/x86\_64-unknown-linux-gnu/4.6.3/../../../../include/c++/4.6.3/bits/cpp\_type\_traits.h" 3

template<>

struct \_\_is\_integer<short>

{

enum { \_\_value = 1 };

typedef \_\_true\_type \_\_type;

};

template<>

struct \_\_is\_integer<unsigned short>

{

enum { \_\_value = 1 };

typedef \_\_true\_type \_\_type;

};

template<>

struct \_\_is\_integer<int>

{

enum { \_\_value = 1 };

typedef \_\_true\_type \_\_type;

};

template<>

struct \_\_is\_integer<unsigned int>

{

enum { \_\_value = 1 };

typedef \_\_true\_type \_\_type;

};

template<>

struct \_\_is\_integer<long>

{

enum { \_\_value = 1 };

typedef \_\_true\_type \_\_type;

};

template<>

struct \_\_is\_integer<unsigned long>

{

enum { \_\_value = 1 };

typedef \_\_true\_type \_\_type;

};

template<>

struct \_\_is\_integer<long long>

{

enum { \_\_value = 1 };

typedef \_\_true\_type \_\_type;

};

template<>

struct \_\_is\_integer<unsigned long long>

{

enum { \_\_value = 1 };

typedef \_\_true\_type \_\_type;

};

template<typename \_Tp>

struct \_\_is\_floating

{

enum { \_\_value = 0 };

typedef \_\_false\_type \_\_type;

};

template<>

struct \_\_is\_floating<float>

{

enum { \_\_value = 1 };

typedef \_\_true\_type \_\_type;

};

template<>

struct \_\_is\_floating<double>

{

enum { \_\_value = 1 };

typedef \_\_true\_type \_\_type;

};

template<>

struct \_\_is\_floating<long double>

{

enum { \_\_value = 1 };

typedef \_\_true\_type \_\_type;

};

template<typename \_Tp>

struct \_\_is\_pointer

{

enum { \_\_value = 0 };

typedef \_\_false\_type \_\_type;

};

template<typename \_Tp>

struct \_\_is\_pointer<\_Tp\*>

{

enum { \_\_value = 1 };

typedef \_\_true\_type \_\_type;

};

template<typename \_Tp>

struct \_\_is\_normal\_iterator

{

enum { \_\_value = 0 };

typedef \_\_false\_type \_\_type;

};

template<typename \_Iterator, typename \_Container>

struct \_\_is\_normal\_iterator< \_\_gnu\_cxx::\_\_normal\_iterator<\_Iterator,

\_Container> >

{

enum { \_\_value = 1 };

typedef \_\_true\_type \_\_type;

};

template<typename \_Tp>

struct \_\_is\_arithmetic

: public \_\_traitor<\_\_is\_integer<\_Tp>, \_\_is\_floating<\_Tp> >

{ };

template<typename \_Tp>

struct \_\_is\_fundamental

: public \_\_traitor<\_\_is\_void<\_Tp>, \_\_is\_arithmetic<\_Tp> >

{ };

template<typename \_Tp>

struct \_\_is\_scalar

: public \_\_traitor<\_\_is\_arithmetic<\_Tp>, \_\_is\_pointer<\_Tp> >

{ };

template<typename \_Tp>

struct \_\_is\_char

{

enum { \_\_value = 0 };

typedef \_\_false\_type \_\_type;

};

template<>

struct \_\_is\_char<char>

{

enum { \_\_value = 1 };

typedef \_\_true\_type \_\_type;

};

template<>

struct \_\_is\_char<wchar\_t>

{

enum { \_\_value = 1 };

typedef \_\_true\_type \_\_type;

};

template<typename \_Tp>

struct \_\_is\_byte

{

enum { \_\_value = 0 };

typedef \_\_false\_type \_\_type;

};

template<>

struct \_\_is\_byte<char>

{

enum { \_\_value = 1 };

typedef \_\_true\_type \_\_type;

};

template<>

struct \_\_is\_byte<signed char>

{

enum { \_\_value = 1 };

typedef \_\_true\_type \_\_type;

};

template<>

struct \_\_is\_byte<unsigned char>

{

enum { \_\_value = 1 };

typedef \_\_true\_type \_\_type;

};

template<typename \_Tp>

struct \_\_is\_move\_iterator

{

enum { \_\_value = 0 };

typedef \_\_false\_type \_\_type;

};

# 422 "/opt/Xilinx/Vivado\_HLS/2017.2/lnx64/tools/gcc/bin/../lib/gcc/x86\_64-unknown-linux-gnu/4.6.3/../../../../include/c++/4.6.3/bits/cpp\_type\_traits.h" 3

}

# 45 "/opt/Xilinx/Vivado\_HLS/2017.2/lnx64/tools/gcc/bin/../lib/gcc/x86\_64-unknown-linux-gnu/4.6.3/../../../../include/c++/4.6.3/cmath" 2 3

# 1 "/opt/Xilinx/Vivado\_HLS/2017.2/lnx64/tools/gcc/bin/../lib/gcc/x86\_64-unknown-linux-gnu/4.6.3/../../../../include/c++/4.6.3/ext/type\_traits.h" 1 3

# 33 "/opt/Xilinx/Vivado\_HLS/2017.2/lnx64/tools/gcc/bin/../lib/gcc/x86\_64-unknown-linux-gnu/4.6.3/../../../../include/c++/4.6.3/ext/type\_traits.h" 3

# 34 "/opt/Xilinx/Vivado\_HLS/2017.2/lnx64/tools/gcc/bin/../lib/gcc/x86\_64-unknown-linux-gnu/4.6.3/../../../../include/c++/4.6.3/ext/type\_traits.h" 3

namespace \_\_gnu\_cxx \_\_attribute\_\_ ((\_\_visibility\_\_ ("default")))

{

template<bool, typename>

struct \_\_enable\_if

{ };

template<typename \_Tp>

struct \_\_enable\_if<true, \_Tp>

{ typedef \_Tp \_\_type; };

template<bool \_Cond, typename \_Iftrue, typename \_Iffalse>

struct \_\_conditional\_type

{ typedef \_Iftrue \_\_type; };

template<typename \_Iftrue, typename \_Iffalse>

struct \_\_conditional\_type<false, \_Iftrue, \_Iffalse>

{ typedef \_Iffalse \_\_type; };

template<typename \_Tp>

struct \_\_add\_unsigned

{

private:

typedef \_\_enable\_if<std::\_\_is\_integer<\_Tp>::\_\_value, \_Tp> \_\_if\_type;

public:

typedef typename \_\_if\_type::\_\_type \_\_type;

};

template<>

struct \_\_add\_unsigned<char>

{ typedef unsigned char \_\_type; };

template<>

struct \_\_add\_unsigned<signed char>

{ typedef unsigned char \_\_type; };

template<>

struct \_\_add\_unsigned<short>

{ typedef unsigned short \_\_type; };

template<>

struct \_\_add\_unsigned<int>

{ typedef unsigned int \_\_type; };

template<>

struct \_\_add\_unsigned<long>

{ typedef unsigned long \_\_type; };

template<>

struct \_\_add\_unsigned<long long>

{ typedef unsigned long long \_\_type; };

template<>

struct \_\_add\_unsigned<bool>;

template<>

struct \_\_add\_unsigned<wchar\_t>;

template<typename \_Tp>

struct \_\_remove\_unsigned

{

private:

typedef \_\_enable\_if<std::\_\_is\_integer<\_Tp>::\_\_value, \_Tp> \_\_if\_type;

public:

typedef typename \_\_if\_type::\_\_type \_\_type;

};

template<>

struct \_\_remove\_unsigned<char>

{ typedef signed char \_\_type; };

template<>

struct \_\_remove\_unsigned<unsigned char>

{ typedef signed char \_\_type; };

template<>

struct \_\_remove\_unsigned<unsigned short>

{ typedef short \_\_type; };

template<>

struct \_\_remove\_unsigned<unsigned int>

{ typedef int \_\_type; };

template<>

struct \_\_remove\_unsigned<unsigned long>

{ typedef long \_\_type; };

template<>

struct \_\_remove\_unsigned<unsigned long long>

{ typedef long long \_\_type; };

template<>

struct \_\_remove\_unsigned<bool>;

template<>

struct \_\_remove\_unsigned<wchar\_t>;

template<typename \_Type>

inline bool

\_\_is\_null\_pointer(\_Type\* \_\_ptr)

{ return \_\_ptr == 0; }

template<typename \_Type>

inline bool

\_\_is\_null\_pointer(\_Type)

{ return false; }

template<typename \_Tp, bool = std::\_\_is\_integer<\_Tp>::\_\_value>

struct \_\_promote

{ typedef double \_\_type; };

template<typename \_Tp>

struct \_\_promote<\_Tp, false>

{ };

template<>

struct \_\_promote<long double>

{ typedef long double \_\_type; };

template<>

struct \_\_promote<double>

{ typedef double \_\_type; };

template<>

struct \_\_promote<float>

{ typedef float \_\_type; };

template<typename \_Tp, typename \_Up,

typename \_Tp2 = typename \_\_promote<\_Tp>::\_\_type,

typename \_Up2 = typename \_\_promote<\_Up>::\_\_type>

struct \_\_promote\_2

{

typedef \_\_typeof\_\_(\_Tp2() + \_Up2()) \_\_type;

};

template<typename \_Tp, typename \_Up, typename \_Vp,

typename \_Tp2 = typename \_\_promote<\_Tp>::\_\_type,

typename \_Up2 = typename \_\_promote<\_Up>::\_\_type,

typename \_Vp2 = typename \_\_promote<\_Vp>::\_\_type>

struct \_\_promote\_3

{

typedef \_\_typeof\_\_(\_Tp2() + \_Up2() + \_Vp2()) \_\_type;

};

template<typename \_Tp, typename \_Up, typename \_Vp, typename \_Wp,

typename \_Tp2 = typename \_\_promote<\_Tp>::\_\_type,

typename \_Up2 = typename \_\_promote<\_Up>::\_\_type,

typename \_Vp2 = typename \_\_promote<\_Vp>::\_\_type,

typename \_Wp2 = typename \_\_promote<\_Wp>::\_\_type>

struct \_\_promote\_4

{

typedef \_\_typeof\_\_(\_Tp2() + \_Up2() + \_Vp2() + \_Wp2()) \_\_type;

};

}

# 46 "/opt/Xilinx/Vivado\_HLS/2017.2/lnx64/tools/gcc/bin/../lib/gcc/x86\_64-unknown-linux-gnu/4.6.3/../../../../include/c++/4.6.3/cmath" 2 3

# 1 "/usr/include/math.h" 1 3 4

# 30 "/usr/include/math.h" 3 4

extern "C" {

# 1 "/usr/include/x86\_64-linux-gnu/bits/huge\_val.h" 1 3 4

# 35 "/usr/include/math.h" 2 3 4

# 1 "/usr/include/x86\_64-linux-gnu/bits/huge\_valf.h" 1 3 4

# 37 "/usr/include/math.h" 2 3 4

# 1 "/usr/include/x86\_64-linux-gnu/bits/huge\_vall.h" 1 3 4

# 38 "/usr/include/math.h" 2 3 4

# 1 "/usr/include/x86\_64-linux-gnu/bits/inf.h" 1 3 4

# 41 "/usr/include/math.h" 2 3 4

# 1 "/usr/include/x86\_64-linux-gnu/bits/nan.h" 1 3 4

# 44 "/usr/include/math.h" 2 3 4

# 1 "/usr/include/x86\_64-linux-gnu/bits/mathdef.h" 1 3 4

# 26 "/usr/include/x86\_64-linux-gnu/bits/mathdef.h" 3 4

# 1 "/usr/include/x86\_64-linux-gnu/bits/wordsize.h" 1 3 4

# 27 "/usr/include/x86\_64-linux-gnu/bits/mathdef.h" 2 3 4

typedef float float\_t;

typedef double double\_t;

# 48 "/usr/include/math.h" 2 3 4

# 71 "/usr/include/math.h" 3 4

# 1 "/usr/include/x86\_64-linux-gnu/bits/mathcalls.h" 1 3 4

# 53 "/usr/include/x86\_64-linux-gnu/bits/mathcalls.h" 3 4

extern double acos (double \_\_x) throw (); extern double \_\_acos (double \_\_x) throw ();

extern double asin (double \_\_x) throw (); extern double \_\_asin (double \_\_x) throw ();

extern double atan (double \_\_x) throw (); extern double \_\_atan (double \_\_x) throw ();

extern double atan2 (double \_\_y, double \_\_x) throw (); extern double \_\_atan2 (double \_\_y, double \_\_x) throw ();

extern double cos (double \_\_x) throw (); extern double \_\_cos (double \_\_x) throw ();

extern double sin (double \_\_x) throw (); extern double \_\_sin (double \_\_x) throw ();

extern double tan (double \_\_x) throw (); extern double \_\_tan (double \_\_x) throw ();

extern double cosh (double \_\_x) throw (); extern double \_\_cosh (double \_\_x) throw ();

extern double sinh (double \_\_x) throw (); extern double \_\_sinh (double \_\_x) throw ();

extern double tanh (double \_\_x) throw (); extern double \_\_tanh (double \_\_x) throw ();

extern void sincos (double \_\_x, double \*\_\_sinx, double \*\_\_cosx) throw (); extern void \_\_sincos (double \_\_x, double \*\_\_sinx, double \*\_\_cosx) throw ()

;

extern double acosh (double \_\_x) throw (); extern double \_\_acosh (double \_\_x) throw ();

extern double asinh (double \_\_x) throw (); extern double \_\_asinh (double \_\_x) throw ();

extern double atanh (double \_\_x) throw (); extern double \_\_atanh (double \_\_x) throw ();

extern double exp (double \_\_x) throw (); extern double \_\_exp (double \_\_x) throw ();

extern double frexp (double \_\_x, int \*\_\_exponent) throw (); extern double \_\_frexp (double \_\_x, int \*\_\_exponent) throw ();

extern double ldexp (double \_\_x, int \_\_exponent) throw (); extern double \_\_ldexp (double \_\_x, int \_\_exponent) throw ();

extern double log (double \_\_x) throw (); extern double \_\_log (double \_\_x) throw ();

extern double log10 (double \_\_x) throw (); extern double \_\_log10 (double \_\_x) throw ();

extern double modf (double \_\_x, double \*\_\_iptr) throw (); extern double \_\_modf (double \_\_x, double \*\_\_iptr) throw ()

\_\_attribute\_\_ ((\_\_nonnull\_\_ (2)));

extern double exp10 (double \_\_x) throw (); extern double \_\_exp10 (double \_\_x) throw ();

extern double pow10 (double \_\_x) throw (); extern double \_\_pow10 (double \_\_x) throw ();

extern double expm1 (double \_\_x) throw (); extern double \_\_expm1 (double \_\_x) throw ();

extern double log1p (double \_\_x) throw (); extern double \_\_log1p (double \_\_x) throw ();

extern double logb (double \_\_x) throw (); extern double \_\_logb (double \_\_x) throw ();

extern double exp2 (double \_\_x) throw (); extern double \_\_exp2 (double \_\_x) throw ();

extern double log2 (double \_\_x) throw (); extern double \_\_log2 (double \_\_x) throw ();

extern double pow (double \_\_x, double \_\_y) throw (); extern double \_\_pow (double \_\_x, double \_\_y) throw ();

extern double sqrt (double \_\_x) throw (); extern double \_\_sqrt (double \_\_x) throw ();

extern double hypot (double \_\_x, double \_\_y) throw (); extern double \_\_hypot (double \_\_x, double \_\_y) throw ();

extern double cbrt (double \_\_x) throw (); extern double \_\_cbrt (double \_\_x) throw ();

extern double ceil (double \_\_x) throw () \_\_attribute\_\_ ((\_\_const\_\_)); extern double \_\_ceil (double \_\_x) throw () \_\_attribute\_\_ ((\_\_const\_\_));

extern double fabs (double \_\_x) throw () \_\_attribute\_\_ ((\_\_const\_\_)); extern double \_\_fabs (double \_\_x) throw () \_\_attribute\_\_ ((\_\_const\_\_));

extern double floor (double \_\_x) throw () \_\_attribute\_\_ ((\_\_const\_\_)); extern double \_\_floor (double \_\_x) throw () \_\_attribute\_\_ ((\_\_const\_\_));

extern double fmod (double \_\_x, double \_\_y) throw (); extern double \_\_fmod (double \_\_x, double \_\_y) throw ();

extern int \_\_isinf (double \_\_value) throw () \_\_attribute\_\_ ((\_\_const\_\_));

extern int \_\_finite (double \_\_value) throw () \_\_attribute\_\_ ((\_\_const\_\_));

extern int isinf (double \_\_value) throw () \_\_attribute\_\_ ((\_\_const\_\_));

extern int finite (double \_\_value) throw () \_\_attribute\_\_ ((\_\_const\_\_));

extern double drem (double \_\_x, double \_\_y) throw (); extern double \_\_drem (double \_\_x, double \_\_y) throw ();

extern double significand (double \_\_x) throw (); extern double \_\_significand (double \_\_x) throw ();

extern double copysign (double \_\_x, double \_\_y) throw () \_\_attribute\_\_ ((\_\_const\_\_)); extern double \_\_copysign (double \_\_x, double \_\_y) throw () \_\_attribute\_\_ ((\_\_const\_\_));

extern double nan (\_\_const char \*\_\_tagb) throw () \_\_attribute\_\_ ((\_\_const\_\_)); extern double \_\_nan (\_\_const char \*\_\_tagb) throw () \_\_attribute\_\_ ((\_\_const\_\_));

extern int \_\_isnan (double \_\_value) throw () \_\_attribute\_\_ ((\_\_const\_\_));

extern int isnan (double \_\_value) throw () \_\_attribute\_\_ ((\_\_const\_\_));

extern double j0 (double) throw (); extern double \_\_j0 (double) throw ();

extern double j1 (double) throw (); extern double \_\_j1 (double) throw ();

extern double jn (int, double) throw (); extern double \_\_jn (int, double) throw ();

extern double y0 (double) throw (); extern double \_\_y0 (double) throw ();

extern double y1 (double) throw (); extern double \_\_y1 (double) throw ();

extern double yn (int, double) throw (); extern double \_\_yn (int, double) throw ();

extern double erf (double) throw (); extern double \_\_erf (double) throw ();

extern double erfc (double) throw (); extern double \_\_erfc (double) throw ();

extern double lgamma (double) throw (); extern double \_\_lgamma (double) throw ();

extern double tgamma (double) throw (); extern double \_\_tgamma (double) throw ();

extern double gamma (double) throw (); extern double \_\_gamma (double) throw ();

extern double lgamma\_r (double, int \*\_\_signgamp) throw (); extern double \_\_lgamma\_r (double, int \*\_\_signgamp) throw ();

extern double rint (double \_\_x) throw (); extern double \_\_rint (double \_\_x) throw ();

extern double nextafter (double \_\_x, double \_\_y) throw () \_\_attribute\_\_ ((\_\_const\_\_)); extern double \_\_nextafter (double \_\_x, double \_\_y) throw () \_\_attribute\_\_ ((\_\_const\_\_));

extern double nexttoward (double \_\_x, long double \_\_y) throw () \_\_attribute\_\_ ((\_\_const\_\_)); extern double \_\_nexttoward (double \_\_x, long double \_\_y) throw () \_\_attribute\_\_ ((\_\_const\_\_));

extern double remainder (double \_\_x, double \_\_y) throw (); extern double \_\_remainder (double \_\_x, double \_\_y) throw ();

extern double scalbn (double \_\_x, int \_\_n) throw (); extern double \_\_scalbn (double \_\_x, int \_\_n) throw ();

extern int ilogb (double \_\_x) throw (); extern int \_\_ilogb (double \_\_x) throw ();

extern double scalbln (double \_\_x, long int \_\_n) throw (); extern double \_\_scalbln (double \_\_x, long int \_\_n) throw ();

extern double nearbyint (double \_\_x) throw (); extern double \_\_nearbyint (double \_\_x) throw ();

extern double round (double \_\_x) throw () \_\_attribute\_\_ ((\_\_const\_\_)); extern double \_\_round (double \_\_x) throw () \_\_attribute\_\_ ((\_\_const\_\_));

extern double trunc (double \_\_x) throw () \_\_attribute\_\_ ((\_\_const\_\_)); extern double \_\_trunc (double \_\_x) throw () \_\_attribute\_\_ ((\_\_const\_\_));

extern double remquo (double \_\_x, double \_\_y, int \*\_\_quo) throw (); extern double \_\_remquo (double \_\_x, double \_\_y, int \*\_\_quo) throw ();

extern long int lrint (double \_\_x) throw (); extern long int \_\_lrint (double \_\_x) throw ();

extern long long int llrint (double \_\_x) throw (); extern long long int \_\_llrint (double \_\_x) throw ();

extern long int lround (double \_\_x) throw (); extern long int \_\_lround (double \_\_x) throw ();

extern long long int llround (double \_\_x) throw (); extern long long int \_\_llround (double \_\_x) throw ();

extern double fdim (double \_\_x, double \_\_y) throw (); extern double \_\_fdim (double \_\_x, double \_\_y) throw ();

extern double fmax (double \_\_x, double \_\_y) throw (); extern double \_\_fmax (double \_\_x, double \_\_y) throw ();

extern double fmin (double \_\_x, double \_\_y) throw (); extern double \_\_fmin (double \_\_x, double \_\_y) throw ();

extern int \_\_fpclassify (double \_\_value) throw ()

\_\_attribute\_\_ ((\_\_const\_\_));

extern int \_\_signbit (double \_\_value) throw ()

\_\_attribute\_\_ ((\_\_const\_\_));

extern double fma (double \_\_x, double \_\_y, double \_\_z) throw (); extern double \_\_fma (double \_\_x, double \_\_y, double \_\_z) throw ();

extern double scalb (double \_\_x, double \_\_n) throw (); extern double \_\_scalb (double \_\_x, double \_\_n) throw ();

# 72 "/usr/include/math.h" 2 3 4

# 94 "/usr/include/math.h" 3 4

# 1 "/usr/include/x86\_64-linux-gnu/bits/mathcalls.h" 1 3 4

# 53 "/usr/include/x86\_64-linux-gnu/bits/mathcalls.h" 3 4

extern float acosf (float \_\_x) throw (); extern float \_\_acosf (float \_\_x) throw ();

extern float asinf (float \_\_x) throw (); extern float \_\_asinf (float \_\_x) throw ();

extern float atanf (float \_\_x) throw (); extern float \_\_atanf (float \_\_x) throw ();

extern float atan2f (float \_\_y, float \_\_x) throw (); extern float \_\_atan2f (float \_\_y, float \_\_x) throw ();

extern float cosf (float \_\_x) throw (); extern float \_\_cosf (float \_\_x) throw ();

extern float sinf (float \_\_x) throw (); extern float \_\_sinf (float \_\_x) throw ();

extern float tanf (float \_\_x) throw (); extern float \_\_tanf (float \_\_x) throw ();

extern float coshf (float \_\_x) throw (); extern float \_\_coshf (float \_\_x) throw ();

extern float sinhf (float \_\_x) throw (); extern float \_\_sinhf (float \_\_x) throw ();

extern float tanhf (float \_\_x) throw (); extern float \_\_tanhf (float \_\_x) throw ();

extern void

sincosf

# 82 "/usr/include/x86\_64-linux-gnu/bits/mathcalls.h" 3 4

(float \_\_x, float \*\_\_sinx, float \*\_\_cosx) throw (); extern void

\_\_sincosf

# 82 "/usr/include/x86\_64-linux-gnu/bits/mathcalls.h" 3 4

(float \_\_x, float \*\_\_sinx, float \*\_\_cosx) throw ()

;

extern float acoshf (float \_\_x) throw (); extern float \_\_acoshf (float \_\_x) throw ();

extern float asinhf (float \_\_x) throw (); extern float \_\_asinhf (float \_\_x) throw ();

extern float atanhf (float \_\_x) throw (); extern float \_\_atanhf (float \_\_x) throw ();

extern float expf (float \_\_x) throw (); extern float \_\_expf (float \_\_x) throw ();

extern float frexpf (float \_\_x, int \*\_\_exponent) throw (); extern float \_\_frexpf (float \_\_x, int \*\_\_exponent) throw ();

extern float ldexpf (float \_\_x, int \_\_exponent) throw (); extern float \_\_ldexpf (float \_\_x, int \_\_exponent) throw ();

extern float logf (float \_\_x) throw (); extern float \_\_logf (float \_\_x) throw ();

extern float log10f (float \_\_x) throw (); extern float \_\_log10f (float \_\_x) throw ();

extern float modff (float \_\_x, float \*\_\_iptr) throw (); extern float \_\_modff (float \_\_x, float \*\_\_iptr) throw ()

\_\_attribute\_\_ ((\_\_nonnull\_\_ (2)));

extern float exp10f (float \_\_x) throw (); extern float \_\_exp10f (float \_\_x) throw ();

extern float pow10f (float \_\_x) throw (); extern float \_\_pow10f (float \_\_x) throw ();

extern float expm1f (float \_\_x) throw (); extern float \_\_expm1f (float \_\_x) throw ();

extern float log1pf (float \_\_x) throw (); extern float \_\_log1pf (float \_\_x) throw ();

extern float logbf (float \_\_x) throw (); extern float \_\_logbf (float \_\_x) throw ();

extern float exp2f (float \_\_x) throw (); extern float \_\_exp2f (float \_\_x) throw ();

extern float log2f (float \_\_x) throw (); extern float \_\_log2f (float \_\_x) throw ();

extern float powf (float \_\_x, float \_\_y) throw (); extern float \_\_powf (float \_\_x, float \_\_y) throw ();

extern float sqrtf (float \_\_x) throw (); extern float \_\_sqrtf (float \_\_x) throw ();

extern float hypotf (float \_\_x, float \_\_y) throw (); extern float \_\_hypotf (float \_\_x, float \_\_y) throw ();

extern float cbrtf (float \_\_x) throw (); extern float \_\_cbrtf (float \_\_x) throw ();

extern float ceilf (float \_\_x) throw () \_\_attribute\_\_ ((\_\_const\_\_)); extern float \_\_ceilf (float \_\_x) throw () \_\_attribute\_\_ ((\_\_const\_\_));

extern float fabsf (float \_\_x) throw () \_\_attribute\_\_ ((\_\_const\_\_)); extern float \_\_fabsf (float \_\_x) throw () \_\_attribute\_\_ ((\_\_const\_\_));

extern float floorf (float \_\_x) throw () \_\_attribute\_\_ ((\_\_const\_\_)); extern float \_\_floorf (float \_\_x) throw () \_\_attribute\_\_ ((\_\_const\_\_));

extern float fmodf (float \_\_x, float \_\_y) throw (); extern float \_\_fmodf (float \_\_x, float \_\_y) throw ();

extern int \_\_isinff (float \_\_value) throw () \_\_attribute\_\_ ((\_\_const\_\_));

extern int \_\_finitef (float \_\_value) throw () \_\_attribute\_\_ ((\_\_const\_\_));

extern int isinff (float \_\_value) throw () \_\_attribute\_\_ ((\_\_const\_\_));

extern int finitef (float \_\_value) throw () \_\_attribute\_\_ ((\_\_const\_\_));

extern float dremf (float \_\_x, float \_\_y) throw (); extern float \_\_dremf (float \_\_x, float \_\_y) throw ();

extern float significandf (float \_\_x) throw (); extern float \_\_significandf (float \_\_x) throw ();

extern float copysignf (float \_\_x, float \_\_y) throw () \_\_attribute\_\_ ((\_\_const\_\_)); extern float \_\_copysignf (float \_\_x, float \_\_y) throw () \_\_attribute\_\_ ((\_\_const\_\_));

extern float nanf (\_\_const char \*\_\_tagb) throw () \_\_attribute\_\_ ((\_\_const\_\_)); extern float \_\_nanf (\_\_const char \*\_\_tagb) throw () \_\_attribute\_\_ ((\_\_const\_\_));

extern int \_\_isnanf (float \_\_value) throw () \_\_attribute\_\_ ((\_\_const\_\_));

extern int isnanf (float \_\_value) throw () \_\_attribute\_\_ ((\_\_const\_\_));

extern float j0f (float) throw (); extern float \_\_j0f (float) throw ();

extern float j1f (float) throw (); extern float \_\_j1f (float) throw ();

extern float jnf (int, float) throw (); extern float \_\_jnf (int, float) throw ();

extern float y0f (float) throw (); extern float \_\_y0f (float) throw ();

extern float y1f (float) throw (); extern float \_\_y1f (float) throw ();

extern float ynf (int, float) throw (); extern float \_\_ynf (int, float) throw ();

extern float erff (float) throw (); extern float \_\_erff (float) throw ();

extern float erfcf (float) throw (); extern float \_\_erfcf (float) throw ();

extern float lgammaf (float) throw (); extern float \_\_lgammaf (float) throw ();

extern float tgammaf (float) throw (); extern float \_\_tgammaf (float) throw ();

extern float gammaf (float) throw (); extern float \_\_gammaf (float) throw ();

extern float lgammaf\_r (float, int \*\_\_signgamp) throw (); extern float \_\_lgammaf\_r (float, int \*\_\_signgamp) throw ();

extern float rintf (float \_\_x) throw (); extern float \_\_rintf (float \_\_x) throw ();

extern float nextafterf (float \_\_x, float \_\_y) throw () \_\_attribute\_\_ ((\_\_const\_\_)); extern float \_\_nextafterf (float \_\_x, float \_\_y) throw () \_\_attribute\_\_ ((\_\_const\_\_));

extern float nexttowardf (float \_\_x, long double \_\_y) throw () \_\_attribute\_\_ ((\_\_const\_\_)); extern float \_\_nexttowardf (float \_\_x, long double \_\_y) throw () \_\_attribute\_\_ ((\_\_const\_\_));

extern float remainderf (float \_\_x, float \_\_y) throw (); extern float \_\_remainderf (float \_\_x, float \_\_y) throw ();

extern float scalbnf (float \_\_x, int \_\_n) throw (); extern float \_\_scalbnf (float \_\_x, int \_\_n) throw ();

extern int ilogbf (float \_\_x) throw (); extern int \_\_ilogbf (float \_\_x) throw ();

extern float scalblnf (float \_\_x, long int \_\_n) throw (); extern float \_\_scalblnf (float \_\_x, long int \_\_n) throw ();

extern float nearbyintf (float \_\_x) throw (); extern float \_\_nearbyintf (float \_\_x) throw ();

extern float roundf (float \_\_x) throw () \_\_attribute\_\_ ((\_\_const\_\_)); extern float \_\_roundf (float \_\_x) throw () \_\_attribute\_\_ ((\_\_const\_\_));

extern float truncf (float \_\_x) throw () \_\_attribute\_\_ ((\_\_const\_\_)); extern float \_\_truncf (float \_\_x) throw () \_\_attribute\_\_ ((\_\_const\_\_));

extern float remquof (float \_\_x, float \_\_y, int \*\_\_quo) throw (); extern float \_\_remquof (float \_\_x, float \_\_y, int \*\_\_quo) throw ();

extern long int lrintf (float \_\_x) throw (); extern long int \_\_lrintf (float \_\_x) throw ();

extern long long int llrintf (float \_\_x) throw (); extern long long int \_\_llrintf (float \_\_x) throw ();

extern long int lroundf (float \_\_x) throw (); extern long int \_\_lroundf (float \_\_x) throw ();

extern long long int llroundf (float \_\_x) throw (); extern long long int \_\_llroundf (float \_\_x) throw ();

extern float fdimf (float \_\_x, float \_\_y) throw (); extern float \_\_fdimf (float \_\_x, float \_\_y) throw ();

extern float fmaxf (float \_\_x, float \_\_y) throw (); extern float \_\_fmaxf (float \_\_x, float \_\_y) throw ();

extern float fminf (float \_\_x, float \_\_y) throw (); extern float \_\_fminf (float \_\_x, float \_\_y) throw ();

extern int \_\_fpclassifyf (float \_\_value) throw ()

\_\_attribute\_\_ ((\_\_const\_\_));

extern int \_\_signbitf (float \_\_value) throw ()

\_\_attribute\_\_ ((\_\_const\_\_));

extern float fmaf (float \_\_x, float \_\_y, float \_\_z) throw (); extern float \_\_fmaf (float \_\_x, float \_\_y, float \_\_z) throw ();

extern float scalbf (float \_\_x, float \_\_n) throw (); extern float \_\_scalbf (float \_\_x, float \_\_n) throw ();

# 95 "/usr/include/math.h" 2 3 4

# 146 "/usr/include/math.h" 3 4

# 1 "/usr/include/x86\_64-linux-gnu/bits/mathcalls.h" 1 3 4

# 53 "/usr/include/x86\_64-linux-gnu/bits/mathcalls.h" 3 4

extern long double acosl (long double \_\_x) throw (); extern long double \_\_acosl (long double \_\_x) throw ();

extern long double asinl (long double \_\_x) throw (); extern long double \_\_asinl (long double \_\_x) throw ();

extern long double atanl (long double \_\_x) throw (); extern long double \_\_atanl (long double \_\_x) throw ();

extern long double atan2l (long double \_\_y, long double \_\_x) throw (); extern long double \_\_atan2l (long double \_\_y, long double \_\_x) throw ();

extern long double cosl (long double \_\_x) throw (); extern long double \_\_cosl (long double \_\_x) throw ();

extern long double sinl (long double \_\_x) throw (); extern long double \_\_sinl (long double \_\_x) throw ();

extern long double tanl (long double \_\_x) throw (); extern long double \_\_tanl (long double \_\_x) throw ();

extern long double coshl (long double \_\_x) throw (); extern long double \_\_coshl (long double \_\_x) throw ();

extern long double sinhl (long double \_\_x) throw (); extern long double \_\_sinhl (long double \_\_x) throw ();

extern long double tanhl (long double \_\_x) throw (); extern long double \_\_tanhl (long double \_\_x) throw ();

extern void

sincosl

# 82 "/usr/include/x86\_64-linux-gnu/bits/mathcalls.h" 3 4

(long double \_\_x, long double \*\_\_sinx, long double \*\_\_cosx) throw (); extern void

\_\_sincosl

# 82 "/usr/include/x86\_64-linux-gnu/bits/mathcalls.h" 3 4

(long double \_\_x, long double \*\_\_sinx, long double \*\_\_cosx) throw ()

;

extern long double acoshl (long double \_\_x) throw (); extern long double \_\_acoshl (long double \_\_x) throw ();

extern long double asinhl (long double \_\_x) throw (); extern long double \_\_asinhl (long double \_\_x) throw ();

extern long double atanhl (long double \_\_x) throw (); extern long double \_\_atanhl (long double \_\_x) throw ();

extern long double expl (long double \_\_x) throw (); extern long double \_\_expl (long double \_\_x) throw ();

extern long double frexpl (long double \_\_x, int \*\_\_exponent) throw (); extern long double \_\_frexpl (long double \_\_x, int \*\_\_exponent) throw ();

extern long double ldexpl (long double \_\_x, int \_\_exponent) throw (); extern long double \_\_ldexpl (long double \_\_x, int \_\_exponent) throw ();

extern long double logl (long double \_\_x) throw (); extern long double \_\_logl (long double \_\_x) throw ();

extern long double log10l (long double \_\_x) throw (); extern long double \_\_log10l (long double \_\_x) throw ();

extern long double modfl (long double \_\_x, long double \*\_\_iptr) throw (); extern long double \_\_modfl (long double \_\_x, long double \*\_\_iptr) throw ()

\_\_attribute\_\_ ((\_\_nonnull\_\_ (2)));

extern long double exp10l (long double \_\_x) throw (); extern long double \_\_exp10l (long double \_\_x) throw ();

extern long double pow10l (long double \_\_x) throw (); extern long double \_\_pow10l (long double \_\_x) throw ();

extern long double expm1l (long double \_\_x) throw (); extern long double \_\_expm1l (long double \_\_x) throw ();

extern long double log1pl (long double \_\_x) throw (); extern long double \_\_log1pl (long double \_\_x) throw ();

extern long double logbl (long double \_\_x) throw (); extern long double \_\_logbl (long double \_\_x) throw ();

extern long double exp2l (long double \_\_x) throw (); extern long double \_\_exp2l (long double \_\_x) throw ();

extern long double log2l (long double \_\_x) throw (); extern long double \_\_log2l (long double \_\_x) throw ();

extern long double powl (long double \_\_x, long double \_\_y) throw (); extern long double \_\_powl (long double \_\_x, long double \_\_y) throw ();

extern long double sqrtl (long double \_\_x) throw (); extern long double \_\_sqrtl (long double \_\_x) throw ();

extern long double hypotl (long double \_\_x, long double \_\_y) throw (); extern long double \_\_hypotl (long double \_\_x, long double \_\_y) throw ();

extern long double cbrtl (long double \_\_x) throw (); extern long double \_\_cbrtl (long double \_\_x) throw ();

extern long double ceill (long double \_\_x) throw () \_\_attribute\_\_ ((\_\_const\_\_)); extern long double \_\_ceill (long double \_\_x) throw () \_\_attribute\_\_ ((\_\_const\_\_));

extern long double fabsl (long double \_\_x) throw () \_\_attribute\_\_ ((\_\_const\_\_)); extern long double \_\_fabsl (long double \_\_x) throw () \_\_attribute\_\_ ((\_\_const\_\_));

extern long double floorl (long double \_\_x) throw () \_\_attribute\_\_ ((\_\_const\_\_)); extern long double \_\_floorl (long double \_\_x) throw () \_\_attribute\_\_ ((\_\_const\_\_));

extern long double fmodl (long double \_\_x, long double \_\_y) throw (); extern long double \_\_fmodl (long double \_\_x, long double \_\_y) throw ();

extern int \_\_isinfl (long double \_\_value) throw () \_\_attribute\_\_ ((\_\_const\_\_));

extern int \_\_finitel (long double \_\_value) throw () \_\_attribute\_\_ ((\_\_const\_\_));

extern int isinfl (long double \_\_value) throw () \_\_attribute\_\_ ((\_\_const\_\_));

extern int finitel (long double \_\_value) throw () \_\_attribute\_\_ ((\_\_const\_\_));

extern long double dreml (long double \_\_x, long double \_\_y) throw (); extern long double \_\_dreml (long double \_\_x, long double \_\_y) throw ();

extern long double significandl (long double \_\_x) throw (); extern long double \_\_significandl (long double \_\_x) throw ();

extern long double copysignl (long double \_\_x, long double \_\_y) throw () \_\_attribute\_\_ ((\_\_const\_\_)); extern long double \_\_copysignl (long double \_\_x, long double \_\_y) throw () \_\_attribute\_\_ ((\_\_const\_\_));

extern long double nanl (\_\_const char \*\_\_tagb) throw () \_\_attribute\_\_ ((\_\_const\_\_)); extern long double \_\_nanl (\_\_const char \*\_\_tagb) throw () \_\_attribute\_\_ ((\_\_const\_\_));

extern int \_\_isnanl (long double \_\_value) throw () \_\_attribute\_\_ ((\_\_const\_\_));

extern int isnanl (long double \_\_value) throw () \_\_attribute\_\_ ((\_\_const\_\_));

extern long double j0l (long double) throw (); extern long double \_\_j0l (long double) throw ();

extern long double j1l (long double) throw (); extern long double \_\_j1l (long double) throw ();

extern long double jnl (int, long double) throw (); extern long double \_\_jnl (int, long double) throw ();

extern long double y0l (long double) throw (); extern long double \_\_y0l (long double) throw ();

extern long double y1l (long double) throw (); extern long double \_\_y1l (long double) throw ();

extern long double ynl (int, long double) throw (); extern long double \_\_ynl (int, long double) throw ();

extern long double erfl (long double) throw (); extern long double \_\_erfl (long double) throw ();

extern long double erfcl (long double) throw (); extern long double \_\_erfcl (long double) throw ();

extern long double lgammal (long double) throw (); extern long double \_\_lgammal (long double) throw ();

extern long double tgammal (long double) throw (); extern long double \_\_tgammal (long double) throw ();

extern long double gammal (long double) throw (); extern long double \_\_gammal (long double) throw ();

extern long double lgammal\_r (long double, int \*\_\_signgamp) throw (); extern long double \_\_lgammal\_r (long double, int \*\_\_signgamp) throw ();

extern long double rintl (long double \_\_x) throw (); extern long double \_\_rintl (long double \_\_x) throw ();

extern long double nextafterl (long double \_\_x, long double \_\_y) throw () \_\_attribute\_\_ ((\_\_const\_\_)); extern long double \_\_nextafterl (long double \_\_x, long double \_\_y) throw () \_\_attribute\_\_ ((\_\_const\_\_));

extern long double nexttowardl (long double \_\_x, long double \_\_y) throw () \_\_attribute\_\_ ((\_\_const\_\_)); extern long double \_\_nexttowardl (long double \_\_x, long double \_\_y) throw () \_\_attribute\_\_ ((\_\_const\_\_));

extern long double remainderl (long double \_\_x, long double \_\_y) throw (); extern long double \_\_remainderl (long double \_\_x, long double \_\_y) throw ();

extern long double scalbnl (long double \_\_x, int \_\_n) throw (); extern long double \_\_scalbnl (long double \_\_x, int \_\_n) throw ();

extern int ilogbl (long double \_\_x) throw (); extern int \_\_ilogbl (long double \_\_x) throw ();

extern long double scalblnl (long double \_\_x, long int \_\_n) throw (); extern long double \_\_scalblnl (long double \_\_x, long int \_\_n) throw ();

extern long double nearbyintl (long double \_\_x) throw (); extern long double \_\_nearbyintl (long double \_\_x) throw ();

extern long double roundl (long double \_\_x) throw () \_\_attribute\_\_ ((\_\_const\_\_)); extern long double \_\_roundl (long double \_\_x) throw () \_\_attribute\_\_ ((\_\_const\_\_));

extern long double truncl (long double \_\_x) throw () \_\_attribute\_\_ ((\_\_const\_\_)); extern long double \_\_truncl (long double \_\_x) throw () \_\_attribute\_\_ ((\_\_const\_\_));

extern long double remquol (long double \_\_x, long double \_\_y, int \*\_\_quo) throw (); extern long double \_\_remquol (long double \_\_x, long double \_\_y, int \*\_\_quo) throw ();

extern long int lrintl (long double \_\_x) throw (); extern long int \_\_lrintl (long double \_\_x) throw ();

extern long long int llrintl (long double \_\_x) throw (); extern long long int \_\_llrintl (long double \_\_x) throw ();

extern long int lroundl (long double \_\_x) throw (); extern long int \_\_lroundl (long double \_\_x) throw ();

extern long long int llroundl (long double \_\_x) throw (); extern long long int \_\_llroundl (long double \_\_x) throw ();

extern long double fdiml (long double \_\_x, long double \_\_y) throw (); extern long double \_\_fdiml (long double \_\_x, long double \_\_y) throw ();

extern long double fmaxl (long double \_\_x, long double \_\_y) throw (); extern long double \_\_fmaxl (long double \_\_x, long double \_\_y) throw ();

extern long double fminl (long double \_\_x, long double \_\_y) throw (); extern long double \_\_fminl (long double \_\_x, long double \_\_y) throw ();

extern int \_\_fpclassifyl (long double \_\_value) throw ()

\_\_attribute\_\_ ((\_\_const\_\_));

extern int \_\_signbitl (long double \_\_value) throw ()

\_\_attribute\_\_ ((\_\_const\_\_));

extern long double fmal (long double \_\_x, long double \_\_y, long double \_\_z) throw (); extern long double \_\_fmal (long double \_\_x, long double \_\_y, long double \_\_z) throw ();

extern long double scalbl (long double \_\_x, long double \_\_n) throw (); extern long double \_\_scalbl (long double \_\_x, long double \_\_n) throw ();

# 147 "/usr/include/math.h" 2 3 4

# 162 "/usr/include/math.h" 3 4

extern int signgam;

# 203 "/usr/include/math.h" 3 4

enum

{

FP\_NAN,

FP\_INFINITE,

FP\_ZERO,

FP\_SUBNORMAL,

FP\_NORMAL

};

# 296 "/usr/include/math.h" 3 4

typedef enum

{

\_IEEE\_ = -1,

\_SVID\_,

\_XOPEN\_,

\_POSIX\_,

\_ISOC\_

} \_LIB\_VERSION\_TYPE;

extern \_LIB\_VERSION\_TYPE \_LIB\_VERSION;

# 319 "/usr/include/math.h" 3 4

struct \_\_exception

{

int type;

char \*name;

double arg1;

double arg2;

double retval;

};

extern int matherr (struct \_\_exception \*\_\_exc) throw ();

# 483 "/usr/include/math.h" 3 4

}

# 47 "/opt/Xilinx/Vivado\_HLS/2017.2/lnx64/tools/gcc/bin/../lib/gcc/x86\_64-unknown-linux-gnu/4.6.3/../../../../include/c++/4.6.3/cmath" 2 3

# 77 "/opt/Xilinx/Vivado\_HLS/2017.2/lnx64/tools/gcc/bin/../lib/gcc/x86\_64-unknown-linux-gnu/4.6.3/../../../../include/c++/4.6.3/cmath" 3

namespace std \_\_attribute\_\_ ((\_\_visibility\_\_ ("default")))

{

inline double

abs(double \_\_x)

{ return \_\_builtin\_fabs(\_\_x); }

inline float

abs(float \_\_x)

{ return \_\_builtin\_fabsf(\_\_x); }

inline long double

abs(long double \_\_x)

{ return \_\_builtin\_fabsl(\_\_x); }

template<typename \_Tp>

inline typename \_\_gnu\_cxx::\_\_enable\_if<\_\_is\_integer<\_Tp>::\_\_value,

double>::\_\_type

abs(\_Tp \_\_x)

{ return \_\_builtin\_fabs(\_\_x); }

using ::acos;

inline float

acos(float \_\_x)

{ return \_\_builtin\_acosf(\_\_x); }

inline long double

acos(long double \_\_x)

{ return \_\_builtin\_acosl(\_\_x); }

template<typename \_Tp>

inline typename \_\_gnu\_cxx::\_\_enable\_if<\_\_is\_integer<\_Tp>::\_\_value,

double>::\_\_type

acos(\_Tp \_\_x)

{ return \_\_builtin\_acos(\_\_x); }

using ::asin;

inline float

asin(float \_\_x)

{ return \_\_builtin\_asinf(\_\_x); }

inline long double

asin(long double \_\_x)

{ return \_\_builtin\_asinl(\_\_x); }

template<typename \_Tp>

inline typename \_\_gnu\_cxx::\_\_enable\_if<\_\_is\_integer<\_Tp>::\_\_value,

double>::\_\_type

asin(\_Tp \_\_x)

{ return \_\_builtin\_asin(\_\_x); }

using ::atan;

inline float

atan(float \_\_x)

{ return \_\_builtin\_atanf(\_\_x); }

inline long double

atan(long double \_\_x)

{ return \_\_builtin\_atanl(\_\_x); }

template<typename \_Tp>

inline typename \_\_gnu\_cxx::\_\_enable\_if<\_\_is\_integer<\_Tp>::\_\_value,

double>::\_\_type

atan(\_Tp \_\_x)

{ return \_\_builtin\_atan(\_\_x); }

using ::atan2;

inline float

atan2(float \_\_y, float \_\_x)

{ return \_\_builtin\_atan2f(\_\_y, \_\_x); }

inline long double

atan2(long double \_\_y, long double \_\_x)

{ return \_\_builtin\_atan2l(\_\_y, \_\_x); }

template<typename \_Tp, typename \_Up>

inline

typename \_\_gnu\_cxx::\_\_promote\_2<\_Tp, \_Up>::\_\_type

atan2(\_Tp \_\_y, \_Up \_\_x)

{

typedef typename \_\_gnu\_cxx::\_\_promote\_2<\_Tp, \_Up>::\_\_type \_\_type;

return atan2(\_\_type(\_\_y), \_\_type(\_\_x));

}

using ::ceil;

inline float

ceil(float \_\_x)

{ return \_\_builtin\_ceilf(\_\_x); }

inline long double

ceil(long double \_\_x)

{ return \_\_builtin\_ceill(\_\_x); }

template<typename \_Tp>

inline typename \_\_gnu\_cxx::\_\_enable\_if<\_\_is\_integer<\_Tp>::\_\_value,

double>::\_\_type

ceil(\_Tp \_\_x)

{ return \_\_builtin\_ceil(\_\_x); }

using ::cos;

inline float

cos(float \_\_x)

{ return \_\_builtin\_cosf(\_\_x); }

inline long double

cos(long double \_\_x)

{ return \_\_builtin\_cosl(\_\_x); }

template<typename \_Tp>

inline typename \_\_gnu\_cxx::\_\_enable\_if<\_\_is\_integer<\_Tp>::\_\_value,

double>::\_\_type

cos(\_Tp \_\_x)

{ return \_\_builtin\_cos(\_\_x); }

using ::cosh;

inline float

cosh(float \_\_x)

{ return \_\_builtin\_coshf(\_\_x); }

inline long double

cosh(long double \_\_x)

{ return \_\_builtin\_coshl(\_\_x); }

template<typename \_Tp>

inline typename \_\_gnu\_cxx::\_\_enable\_if<\_\_is\_integer<\_Tp>::\_\_value,

double>::\_\_type

cosh(\_Tp \_\_x)

{ return \_\_builtin\_cosh(\_\_x); }

using ::exp;

inline float

exp(float \_\_x)

{ return \_\_builtin\_expf(\_\_x); }

inline long double

exp(long double \_\_x)

{ return \_\_builtin\_expl(\_\_x); }

template<typename \_Tp>

inline typename \_\_gnu\_cxx::\_\_enable\_if<\_\_is\_integer<\_Tp>::\_\_value,

double>::\_\_type

exp(\_Tp \_\_x)

{ return \_\_builtin\_exp(\_\_x); }

using ::fabs;

inline float

fabs(float \_\_x)

{ return \_\_builtin\_fabsf(\_\_x); }

inline long double

fabs(long double \_\_x)

{ return \_\_builtin\_fabsl(\_\_x); }

template<typename \_Tp>

inline typename \_\_gnu\_cxx::\_\_enable\_if<\_\_is\_integer<\_Tp>::\_\_value,

double>::\_\_type

fabs(\_Tp \_\_x)

{ return \_\_builtin\_fabs(\_\_x); }

using ::floor;

inline float

floor(float \_\_x)

{ return \_\_builtin\_floorf(\_\_x); }

inline long double

floor(long double \_\_x)

{ return \_\_builtin\_floorl(\_\_x); }

template<typename \_Tp>

inline typename \_\_gnu\_cxx::\_\_enable\_if<\_\_is\_integer<\_Tp>::\_\_value,

double>::\_\_type

floor(\_Tp \_\_x)

{ return \_\_builtin\_floor(\_\_x); }

using ::fmod;

inline float

fmod(float \_\_x, float \_\_y)

{ return \_\_builtin\_fmodf(\_\_x, \_\_y); }

inline long double

fmod(long double \_\_x, long double \_\_y)

{ return \_\_builtin\_fmodl(\_\_x, \_\_y); }

using ::frexp;

inline float

frexp(float \_\_x, int\* \_\_exp)

{ return \_\_builtin\_frexpf(\_\_x, \_\_exp); }

inline long double

frexp(long double \_\_x, int\* \_\_exp)

{ return \_\_builtin\_frexpl(\_\_x, \_\_exp); }

template<typename \_Tp>

inline typename \_\_gnu\_cxx::\_\_enable\_if<\_\_is\_integer<\_Tp>::\_\_value,

double>::\_\_type

frexp(\_Tp \_\_x, int\* \_\_exp)

{ return \_\_builtin\_frexp(\_\_x, \_\_exp); }

using ::ldexp;

inline float

ldexp(float \_\_x, int \_\_exp)

{ return \_\_builtin\_ldexpf(\_\_x, \_\_exp); }

inline long double

ldexp(long double \_\_x, int \_\_exp)

{ return \_\_builtin\_ldexpl(\_\_x, \_\_exp); }

template<typename \_Tp>

inline typename \_\_gnu\_cxx::\_\_enable\_if<\_\_is\_integer<\_Tp>::\_\_value,

double>::\_\_type

ldexp(\_Tp \_\_x, int \_\_exp)

{ return \_\_builtin\_ldexp(\_\_x, \_\_exp); }

using ::log;

inline float

log(float \_\_x)

{ return \_\_builtin\_logf(\_\_x); }

inline long double

log(long double \_\_x)

{ return \_\_builtin\_logl(\_\_x); }

template<typename \_Tp>

inline typename \_\_gnu\_cxx::\_\_enable\_if<\_\_is\_integer<\_Tp>::\_\_value,

double>::\_\_type

log(\_Tp \_\_x)

{ return \_\_builtin\_log(\_\_x); }

using ::log10;

inline float

log10(float \_\_x)

{ return \_\_builtin\_log10f(\_\_x); }

inline long double

log10(long double \_\_x)

{ return \_\_builtin\_log10l(\_\_x); }

template<typename \_Tp>

inline typename \_\_gnu\_cxx::\_\_enable\_if<\_\_is\_integer<\_Tp>::\_\_value,

double>::\_\_type

log10(\_Tp \_\_x)

{ return \_\_builtin\_log10(\_\_x); }

using ::modf;

inline float

modf(float \_\_x, float\* \_\_iptr)

{ return \_\_builtin\_modff(\_\_x, \_\_iptr); }

inline long double

modf(long double \_\_x, long double\* \_\_iptr)

{ return \_\_builtin\_modfl(\_\_x, \_\_iptr); }

using ::pow;

inline float

pow(float \_\_x, float \_\_y)

{ return \_\_builtin\_powf(\_\_x, \_\_y); }

inline long double

pow(long double \_\_x, long double \_\_y)

{ return \_\_builtin\_powl(\_\_x, \_\_y); }

inline double

pow(double \_\_x, int \_\_i)

{ return \_\_builtin\_powi(\_\_x, \_\_i); }

inline float

pow(float \_\_x, int \_\_n)

{ return \_\_builtin\_powif(\_\_x, \_\_n); }

inline long double

pow(long double \_\_x, int \_\_n)

{ return \_\_builtin\_powil(\_\_x, \_\_n); }

template<typename \_Tp, typename \_Up>

inline

typename \_\_gnu\_cxx::\_\_promote\_2<\_Tp, \_Up>::\_\_type

pow(\_Tp \_\_x, \_Up \_\_y)

{

typedef typename \_\_gnu\_cxx::\_\_promote\_2<\_Tp, \_Up>::\_\_type \_\_type;

return pow(\_\_type(\_\_x), \_\_type(\_\_y));

}

using ::sin;

inline float

sin(float \_\_x)

{ return \_\_builtin\_sinf(\_\_x); }

inline long double

sin(long double \_\_x)

{ return \_\_builtin\_sinl(\_\_x); }

template<typename \_Tp>

inline typename \_\_gnu\_cxx::\_\_enable\_if<\_\_is\_integer<\_Tp>::\_\_value,

double>::\_\_type

sin(\_Tp \_\_x)

{ return \_\_builtin\_sin(\_\_x); }

using ::sinh;

inline float

sinh(float \_\_x)

{ return \_\_builtin\_sinhf(\_\_x); }

inline long double

sinh(long double \_\_x)

{ return \_\_builtin\_sinhl(\_\_x); }

template<typename \_Tp>

inline typename \_\_gnu\_cxx::\_\_enable\_if<\_\_is\_integer<\_Tp>::\_\_value,

double>::\_\_type

sinh(\_Tp \_\_x)

{ return \_\_builtin\_sinh(\_\_x); }

using ::sqrt;

inline float

sqrt(float \_\_x)

{ return \_\_builtin\_sqrtf(\_\_x); }

inline long double

sqrt(long double \_\_x)

{ return \_\_builtin\_sqrtl(\_\_x); }

template<typename \_Tp>

inline typename \_\_gnu\_cxx::\_\_enable\_if<\_\_is\_integer<\_Tp>::\_\_value,

double>::\_\_type

sqrt(\_Tp \_\_x)

{ return \_\_builtin\_sqrt(\_\_x); }

using ::tan;

inline float

tan(float \_\_x)

{ return \_\_builtin\_tanf(\_\_x); }

inline long double

tan(long double \_\_x)

{ return \_\_builtin\_tanl(\_\_x); }

template<typename \_Tp>

inline typename \_\_gnu\_cxx::\_\_enable\_if<\_\_is\_integer<\_Tp>::\_\_value,

double>::\_\_type

tan(\_Tp \_\_x)

{ return \_\_builtin\_tan(\_\_x); }

using ::tanh;

inline float

tanh(float \_\_x)

{ return \_\_builtin\_tanhf(\_\_x); }

inline long double

tanh(long double \_\_x)

{ return \_\_builtin\_tanhl(\_\_x); }

template<typename \_Tp>

inline typename \_\_gnu\_cxx::\_\_enable\_if<\_\_is\_integer<\_Tp>::\_\_value,

double>::\_\_type

tanh(\_Tp \_\_x)

{ return \_\_builtin\_tanh(\_\_x); }

}

# 481 "/opt/Xilinx/Vivado\_HLS/2017.2/lnx64/tools/gcc/bin/../lib/gcc/x86\_64-unknown-linux-gnu/4.6.3/../../../../include/c++/4.6.3/cmath" 3

namespace std \_\_attribute\_\_ ((\_\_visibility\_\_ ("default")))

{

# 731 "/opt/Xilinx/Vivado\_HLS/2017.2/lnx64/tools/gcc/bin/../lib/gcc/x86\_64-unknown-linux-gnu/4.6.3/../../../../include/c++/4.6.3/cmath" 3

template<typename \_Tp>

inline typename \_\_gnu\_cxx::\_\_enable\_if<\_\_is\_arithmetic<\_Tp>::\_\_value,

int>::\_\_type

fpclassify(\_Tp \_\_f)

{

typedef typename \_\_gnu\_cxx::\_\_promote<\_Tp>::\_\_type \_\_type;

return \_\_builtin\_fpclassify(FP\_NAN, FP\_INFINITE, FP\_NORMAL,

FP\_SUBNORMAL, FP\_ZERO, \_\_type(\_\_f));

}

template<typename \_Tp>

inline typename \_\_gnu\_cxx::\_\_enable\_if<\_\_is\_arithmetic<\_Tp>::\_\_value,

int>::\_\_type

isfinite(\_Tp \_\_f)

{

typedef typename \_\_gnu\_cxx::\_\_promote<\_Tp>::\_\_type \_\_type;

return \_\_builtin\_isfinite(\_\_type(\_\_f));

}

template<typename \_Tp>

inline typename \_\_gnu\_cxx::\_\_enable\_if<\_\_is\_arithmetic<\_Tp>::\_\_value,

int>::\_\_type

isinf(\_Tp \_\_f)

{

typedef typename \_\_gnu\_cxx::\_\_promote<\_Tp>::\_\_type \_\_type;

return \_\_builtin\_isinf(\_\_type(\_\_f));

}

template<typename \_Tp>

inline typename \_\_gnu\_cxx::\_\_enable\_if<\_\_is\_arithmetic<\_Tp>::\_\_value,

int>::\_\_type

isnan(\_Tp \_\_f)

{

typedef typename \_\_gnu\_cxx::\_\_promote<\_Tp>::\_\_type \_\_type;

return \_\_builtin\_isnan(\_\_type(\_\_f));

}

template<typename \_Tp>

inline typename \_\_gnu\_cxx::\_\_enable\_if<\_\_is\_arithmetic<\_Tp>::\_\_value,

int>::\_\_type

isnormal(\_Tp \_\_f)

{

typedef typename \_\_gnu\_cxx::\_\_promote<\_Tp>::\_\_type \_\_type;

return \_\_builtin\_isnormal(\_\_type(\_\_f));

}

template<typename \_Tp>

inline typename \_\_gnu\_cxx::\_\_enable\_if<\_\_is\_arithmetic<\_Tp>::\_\_value,

int>::\_\_type

signbit(\_Tp \_\_f)

{

typedef typename \_\_gnu\_cxx::\_\_promote<\_Tp>::\_\_type \_\_type;

return \_\_builtin\_signbit(\_\_type(\_\_f));

}

template<typename \_Tp>

inline typename \_\_gnu\_cxx::\_\_enable\_if<\_\_is\_arithmetic<\_Tp>::\_\_value,

int>::\_\_type

isgreater(\_Tp \_\_f1, \_Tp \_\_f2)

{

typedef typename \_\_gnu\_cxx::\_\_promote<\_Tp>::\_\_type \_\_type;

return \_\_builtin\_isgreater(\_\_type(\_\_f1), \_\_type(\_\_f2));

}

template<typename \_Tp>

inline typename \_\_gnu\_cxx::\_\_enable\_if<\_\_is\_arithmetic<\_Tp>::\_\_value,

int>::\_\_type

isgreaterequal(\_Tp \_\_f1, \_Tp \_\_f2)

{

typedef typename \_\_gnu\_cxx::\_\_promote<\_Tp>::\_\_type \_\_type;

return \_\_builtin\_isgreaterequal(\_\_type(\_\_f1), \_\_type(\_\_f2));

}

template<typename \_Tp>

inline typename \_\_gnu\_cxx::\_\_enable\_if<\_\_is\_arithmetic<\_Tp>::\_\_value,

int>::\_\_type

isless(\_Tp \_\_f1, \_Tp \_\_f2)

{

typedef typename \_\_gnu\_cxx::\_\_promote<\_Tp>::\_\_type \_\_type;

return \_\_builtin\_isless(\_\_type(\_\_f1), \_\_type(\_\_f2));

}

template<typename \_Tp>

inline typename \_\_gnu\_cxx::\_\_enable\_if<\_\_is\_arithmetic<\_Tp>::\_\_value,

int>::\_\_type

islessequal(\_Tp \_\_f1, \_Tp \_\_f2)

{

typedef typename \_\_gnu\_cxx::\_\_promote<\_Tp>::\_\_type \_\_type;

return \_\_builtin\_islessequal(\_\_type(\_\_f1), \_\_type(\_\_f2));

}

template<typename \_Tp>

inline typename \_\_gnu\_cxx::\_\_enable\_if<\_\_is\_arithmetic<\_Tp>::\_\_value,

int>::\_\_type

islessgreater(\_Tp \_\_f1, \_Tp \_\_f2)

{

typedef typename \_\_gnu\_cxx::\_\_promote<\_Tp>::\_\_type \_\_type;

return \_\_builtin\_islessgreater(\_\_type(\_\_f1), \_\_type(\_\_f2));

}

template<typename \_Tp>

inline typename \_\_gnu\_cxx::\_\_enable\_if<\_\_is\_arithmetic<\_Tp>::\_\_value,

int>::\_\_type

isunordered(\_Tp \_\_f1, \_Tp \_\_f2)

{

typedef typename \_\_gnu\_cxx::\_\_promote<\_Tp>::\_\_type \_\_type;

return \_\_builtin\_isunordered(\_\_type(\_\_f1), \_\_type(\_\_f2));

}

}

# 70 "/home/drsatya/Desktop/lab1/matrixmul.h" 2

using namespace std;

# 80 "/home/drsatya/Desktop/lab1/matrixmul.h"

typedef char mat\_a\_t;

typedef char mat\_b\_t;

typedef short result\_t;

void matrixmul(

mat\_a\_t a[3][3],

mat\_b\_t b[3][3],

result\_t res[3][3]);

# 3 "/home/drsatya/Desktop/lab1/matrixmul.cpp" 2

void matrixmul(

mat\_a\_t a[3][3],

mat\_b\_t b[3][3],

result\_t res[3][3])

{

Row: for(int i = 0; i < 3; i++) {

Col: for(int j = 0; j < 3; j++) {

res[i][j] = 0;

Product: for(int k = 0; k < 3; k++) {

res[i][j] += a[i][k] \* b[k][j];

}

}

}

}