List of Maths Library Functions

Notes
1. Functions found in the C-math library are indicated **thus**.
2. Integer functions and integer results are highlighted **thus**.
3. In the descriptions, X and Y indicate real numbers, A, B, N integer values.
4. Violation of preconditions on input arguments will result in an error message.
5. Post-conditions, where applicable, are stated.

abs : number --> number   **fabs** in C

Purpose: returns $Y = |X|$.   Post-condition: $Y \geq 0$

acos : number --> number

Purpose: returns Y, the arcsine of X. Precondition: $-1 \leq X \leq +1$. Post-condition: $0 \leq Y \leq \pi$

asin : number --> number

Purpose: returns Y, the arcsine of X. Precondition: $-1 \leq X \leq +1$. Post-condition: $-\pi/2 \leq Y \leq +\pi/2$

atan : number --> number

Purpose: returns Y, the arctangent of X.   Post-condition: $-\pi/2 < Y < +\pi/2$

atan2 : number --> number --> number

Purpose: returns Y, the two argument arctangent.   Post-condition: $-\pi < Y \leq +\pi$

ceiling : number --> number   **ceil** in C

Purpose: returns $\lceil X \rceil$.

cos : number --> number

Purpose: returns Y, the cosine of X.   Post-condition: $-1 \leq Y \leq 1$

cosh : number --> number

Purpose: returns the hyperbolic cosine of X.

degs->dms : number --> (list number)

Purpose: converts fractional degrees to (a list) [degrees minutes seconds].

degs->rad : number --> number

Purpose: converts fractional degrees to radians.

div : number --> number --> number

Purpose: returns the floored integer quotient of A and B.

div-eucl : number --> number --> number

Purpose: returns the Eulidean integer quotient of A and B.

divisible-by? : number --> number --> boolean

Purpose: tests if A is divisible by B. Precondition: both arguments must be positive integers!

dms->degs : (list number) --> number

Purpose: converts a valid list [degrees minutes seconds] to fractional degrees.

```
even? :  number --> boolean
```
Purpose: tests if the argument is an even integer.

```
exp : number --> number
```
Purpose: returns the value of e^X.

```
expt : number --> number --> number          pow in C
```
Purpose: returns the value of X^Y, or an error message if not defined

```
floor :  number --> number
```
Purpose: returns $\lfloor X \rfloor$.

```
fmod : number --> number --> number
```
Purpose: returns the floating-point modulus of X modulo Y.

```
frac-part :  number --> number
```
Purpose: returns the fractional part of X.

```
frexp : number --> (number * number)
```
Purpose: 'free-the-exponent' returns a pair consisting of mantissa and exponent of X.

```
gcd : number --> . . . --> number
```
Purpose: returns the greatest common divisor of its integer arguments (result always ≥ 0).

```
int-part :  number --> number
```
Purpose: returns the integer part of X. Same as **trunc**.

```
lcm : number --> . . . --> number
```
Purpose: returns the least common multiple of its integer arguments (result always ≥ 0).

```
ldexp : number --> number --> number
```
Purpose: 'load-the-exponent' returns a number with specified mantissa and exponent.

```
log : number --> number
```
Purpose: returns the natural logarithm of X. Precondition: $X > 0$.

```
log2 : number --> number
```
Purpose: returns the base-2 logarithm of X. Precondition: $X > 0$.

```
log10 : number --> number
```
Purpose: returns the base-10 logarithm of X. Precondition: $X > 0$.

```
log' : number --> number --> number
```
Purpose: returns the base-Y logarithm of X. Precondition: $X > 0, Y > 0$.

```
max : number --> . . . --> number
```
Purpose: returns the largest of its arguments.

```
min : number --> . . . --> number
```
Purpose: returns the smallest of its arguments.

`mod :  number --> number --> number`

Purpose: returns the remainder on dividing A by B. The remainder has same sign as B.

`modf :  number --> (number * number)`

Purpose: returns the pair consisting of integer part and fractional part of X.

`natural? :  number --> boolean`

Purpose: tests if X is a natural number.

`negative? :  number --> boolean`

Purpose: tests if X is a negative number.

`odd? :  number --> boolean`

Purpose: tests if X is an odd integer.

`positive? :  number --> boolean`

Purpose: tests if X is a positive number.

`power : number --> number --> number`

Purpose: returns the value of X^N, for integer N.

`prime? :  number --> boolean`

Purpose: tests if N is a prime number.

`rad->degs : number --> number`

Purpose: converts an angle from radians to fractional degrees.

`rem :  number --> number --> number`

Purpose: returns the remainder on dividing A by B. The remainder has same sign as A.

`round :  number --> number`   (same as **maths-round0**)

Purpose: rounds the argument to the nearest integer.

`round :  number --> number --> number`   (same as **maths-round'**)

Purpose: rounds the argument to a specified number of places.

`sign :  number --> number`

Purpose: returns the sign of X.

`sin : number --> number`

Purpose: returns Y, the sine of X.   Post-condition: $-1 \leq Y \leq 1$

`sinh : number --> number`

Purpose: returns the hyperbolic sine of X.

`square : number --> number`

Purpose: returns the value of X^2.

`sqrt : number --> number`

Purpose: returns the value of $\sqrt{X}$. Precondition: $X \geq 0$

tan : number --> number

Purpose: returns the tangent of X.

tanh : number --> number

Purpose: returns the hyperbolic tangent of X.

trunc : number --> number

Purpose: returns the integer part of X. Same as **int-part**.

trunc-div : number --> number --> number

Purpose: returns the truncated integer quotient of A and B.

% : number --> number --> number

Purpose: returns the remainder on dividing A by B. The remainder is always ≥ 0.

/% : number --> number --> (number * number)

Purpose: returns the pair consisting of the Euclidean integer quotient and remainder.

/mod : number --> number --> (number * number)

Purpose: returns the pair consisting of the floored integer quotient and remainder.

/rem : number --> number --> (number * number)

Purpose: returns the pair consisting of the truncated integer quotient and remainder.