

**mathieeedoubbas**

<b>COLLABORATORS</b>
----------------------

	<i>TITLE :</i> mathieeedoubbas		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		July 18, 2024	

<b>REVISION HISTORY</b>
-------------------------

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>mathieeedoubbas</b>	<b>1</b>
1.1	mathieeedoubbas.doc . . . . .	1
1.2	mathieeedoubbas.library/IEEEDPAbs . . . . .	1
1.3	mathieeedoubbas.library/IEEEDPAdd . . . . .	1
1.4	mathieeedoubbas.library/IEEEDPCeil . . . . .	2
1.5	mathieeedoubbas.library/IEEEDPComp . . . . .	2
1.6	mathieeedoubbas.library/IEEEDPDiv . . . . .	3
1.7	mathieeedoubbas.library/IEEEDPFix . . . . .	4
1.8	mathieeedoubbas.library/IEEEDPFloor . . . . .	4
1.9	mathieeedoubbas.library/IEEEDPFIt . . . . .	5
1.10	mathieeedoubbas.library/IEEEDPMul . . . . .	5
1.11	mathieeedoubbas.library/IEEEDPNeg . . . . .	6
1.12	mathieeedoubbas.library/IEEEDPSub . . . . .	6
1.13	mathieeedoubbas.library/IEEEDPtst . . . . .	7

## Chapter 1

# mathieeedoubbas

### 1.1 mathieeedoubbas.doc

IEEEDPAbs()	IEEEDPComp()	IEEEDPFloor()	IEEEDPNeg()
IEEEDPAdd()	IEEEDPDiv()	IEEEDPFlt()	IEEEDPSub()
IEEEDPCeil()	IEEEDPFix()	IEEEDPMul()	IEEEDPtst()

### 1.2 mathieeedoubbas.library/IEEEDPAbs

#### NAME

IEEEDPAbs -- compute absolute value of IEEE double precision argument

#### SYNOPSIS

```
    x    = IEEEDPAbs( y );  
    d0/d1      d0/d1
```

```
double  x,y;
```

#### FUNCTION

Take the absolute value of argument y and return it to caller.

#### INPUTS

y -- IEEE double precision floating point value

#### RESULT

x -- IEEE double precision floating point value

#### BUGS

#### SEE ALSO

### 1.3 mathieeedoubbas.library/IEEEDPAdd

#### NAME

IEEEDPAdd -- add one double precision IEEE number to another

## SYNOPSIS

```

    x    = IEEEEDPAdd( y , z );
    d0/d1      d0/d1 d2/d3

```

```

double x,y,z;

```

## FUNCTION

Compute  $x = y + z$  in IEEE double precision.

## INPUTS

```

y -- IEEE double precision floating point value
z -- IEEE double precision floating point value

```

## RESULT

```

x -- IEEE double precision floating point value

```

## BUGS

## SEE ALSO

IEEEEDPSub

## 1.4 mathieeedoubbas.library/IEEEDPCeil

## NAME

IEEEDPCeil -- compute Ceil function of IEEE double precision number

## SYNOPSIS

```

    x    = IEEEDPCeil( y );
    d0/d1      d0/d1

```

```

double x,y;

```

## FUNCTION

Calculate the least integer greater than or equal to  $x$  and return it.  
 This value may have more than 32 bits of significance.  
 This identity is true.  $\text{Ceil}(x) = -\text{Floor}(-x)$ .

## INPUTS

```

y -- IEEE double precision floating point value

```

## RESULT

```

x -- IEEE double precision floating point value

```

## BUGS

## SEE ALSO

IEEEDPFloor

## 1.5 mathieeedoubbas.library/IEEEDPComp

## NAME

IEEEDPComp -- compare two double precision floating point numbers

## SYNOPSIS

```

    c    = IEEEEDPCmp( y , z );
    d0    d0/d1 d2/d3

```

```

double y,z;
long    c;

```

## FUNCTION

Compare y with z. Set the condition codes for less, greater, or equal. Set return value c to -1 if y<z, or +1 if y>z, or 0 if y == z.

## INPUTS

```

y -- IEEE double precision floating point value
z -- IEEE double precision floating point value

```

## RESULT

```

c = 1    cc = gt      for (y > z)
c = 0    cc = eq      for (y == z)
c = -1   cc = lt      for (y < z)

```

## BUGS

## SEE ALSO

## 1.6 mathieedoubbas.library/IEEEDPDiv

## NAME

IEEEDPDiv -- divide one double precision IEEE by another

## SYNOPSIS

```

    x    = IEEEDPDiv( y , z );
    d0/d1    d0/d1 d2/d3

```

```

double x,y,z;

```

## FUNCTION

Compute x = y / z in IEEE double precision.

## INPUTS

```

y -- IEEE double precision floating point value
z -- IEEE double precision floating point value

```

## RESULT

```

x -- IEEE double precision floating point value

```

## BUGS

## SEE ALSO

IEEEDPMul

## 1.7 mathieedoubbas.library/IEEEDPFix

### NAME

IEEEDPFix -- convert IEEE double float to integer

### SYNOPSIS

```
x    = IEEEDPFix( y );  
d0      d0/d1
```

```
long    x;  
double  y;
```

### FUNCTION

Convert IEEE double precision argument to a 32 bit signed integer and return result.

### INPUTS

y -- IEEE double precision floating point value

### RESULT

```
if no overflow occurred then return  
    x -- 32 bit signed integer  
if overflow return largest +- integer  
    For round to zero
```

### BUGS

### SEE ALSO

IEEEDPFlt

## 1.8 mathieedoubbas.library/IEEEDPFloor

### NAME

IEEEDPFloor -- compute Floor function of IEEE double precision number

### SYNOPSIS

```
x    = IEEEDPFloor( y );  
d0/d1      d0/d1
```

```
double  x,y;
```

### FUNCTION

Calculate the largest integer less than or equal to x and return it. This value may have more than 32 bits of significance.

### INPUTS

y -- IEEE double precision floating point value

### RESULT

x -- IEEE double precision floating point value

### BUGS

### SEE ALSO

---

IEEEDPCeil

## 1.9 mathieedoubbas.library/IEEEDPFIt

### NAME

IEEEDPFIt -- convert integer to IEEE double precision number

### SYNOPSIS

```

    x    = IEEEDPFIt( y );
    d0/d1      d0

```

```

double  x;
long    y;

```

### FUNCTION

Convert a signed 32 bit value to a double precision IEEE value and return it in d0/d1. No exceptions can occur with this function.

### INPUTS

y -- 32 bit integer in d0

### RESULT

x is a 64 bit double precision IEEE value

### BUGS

### SEE ALSO

IEEEDPFix

## 1.10 mathieedoubbas.library/IEEEDPMul

### NAME

IEEEDPMul -- multiply one double precision IEEE number by another

### SYNOPSIS

```

    x    = IEEEDPMul( y , z );
    d0/d1      d0/d1 d2/d3

```

```

double  x,y,z;

```

### FUNCTION

Compute  $x = y * z$  in IEEE double precision.

### INPUTS

y -- IEEE double precision floating point value  
 z -- IEEE double precision floating point value

### RESULT

x -- IEEE double precision floating point value

### BUGS



SEE ALSO  
IEEEEDPDiv

## 1.11 mathieedoubbas.library/IEEEDPNeg

### NAME

IEEEDPNeg -- compute negative value of IEEE double precision number

### SYNOPSIS

```
    x    = IEEEDPNeg( y );
d0/d1      d0/d1

double  x,y;
```

### FUNCTION

Invert the sign of argument y and return it to caller.

### INPUTS

y - IEEE double precision floating point value

### RESULT

x - IEEE double precision floating point value

### BUGS

SEE ALSO

## 1.12 mathieedoubbas.library/IEEEDPSub

### NAME

IEEEDPSub -- subtract one double precision IEEE number from another

### SYNOPSIS

```
    x    = IEEEDPSub( y , z );
d0/d1      d0/d1 d2/d3

double  x,y,z;
```

### FUNCTION

Compute  $x = y - z$  in IEEE double precision.

### INPUTS

y -- IEEE double precision floating point value  
z -- IEEE double precision floating point value

### RESULT

x -- IEEE double precision floating point value

### BUGS

SEE ALSO

---

IEEEEDPAdd

## 1.13 mathieedoubbas.library/IEEEEDPTst

### NAME

IEEEEDPTst -- compare IEEE double precision value to 0.0

### SYNOPSIS

```
c    = IEEEEDPTst( y );  
d0           d0/d1
```

```
double y;  
long   c;
```

### FUNCTION

Compare y to 0.0, set the condition codes for less than, greater than, or equal to 0.0. Set the return value c to -1 if less than, to +1 if greater than, or 0 if equal to 0.0.

### INPUTS

y -- IEEE double precision floating point value

### RESULT

c = 1	cc = gt	for (y > 0.0)
c = 0	cc = eq	for (y == 0.0)
c = -1	cc = lt	for (y < 0.0)

### BUGS

### SEE ALSO