Scientific Calculator

Generated by Doxygen 1.8.16

1 Main Page	1
2 Data Structure Index	3
2.1 Data Structures	3
3 File Index	5
3.1 File List	5
4 Data Structure Documentation	7
4.1 Complex Struct Reference	7
4.1.1 Detailed Description	7
4.1.2 Field Documentation	7
4.1.2.1 imag	7
4.1.2.2 real	7
4.2 Matrix Struct Reference	8
4.2.1 Detailed Description	8
4.2.2 Field Documentation	8
4.2.2.1 n_columns	8
4.2.2.2 n_rows	8
4.2.2.3 values	8
5 File Documentation	9
5.1 calculator.c File Reference	9
5.1.1 Detailed Description	9
5.2 math_library.c File Reference	10
5.2.1 Detailed Description	10
5.2.2 Function Documentation	11
5.2.2.1 add_matrices()	11
5.2.2.2 create_matrix()	12
5.2.2.3 factorial()	13
5.2.2.4 multiply_matrices()	14
5.2.2.5 poly_discriminant()	15
5.2.2.6 print_matrix()	16
5.2.2.7 read_matrix()	16
5.2.2.8 transpose_matrix()	17
5.3 math_library.h File Reference	18
5.3.1 Detailed Description	19
5.3.2 Function Documentation	19
5.3.2.1 add_matrices()	19
5.3.2.2 create_matrix()	20
5.3.2.3 factorial()	21
5.3.2.4 multiply_matrices()	22
5.3.2.5 poly_discriminant()	23
5.3.2.6 print_matrix()	24
· · · · · · · · · · · · · · · · · · ·	

5.3.2.7 read_matrix()	24
5.3.2.8 sine()	25
5.3.2.9 transpose_matrix()	25

# **Chapter 1**

# Main Page

Computer scientists are often required to conduct complex mathematical computations. This calculator is a program implementing functions to automate some of these computations.

#### **List of Implemented Functions**

- · Transpose matrix.
- · Add two matrices.
- · Multiply two matrices.
- · Compute roots of quadratic polynomial.
- · Compute factorial.

# **Program Execution**

To execute this program cmake is required to be installed. Open a terminal and change into the directory of the scientific\_calculator. Afterwards run the following commands.

```
cmake --build cmake-build-debug --target scientific_calculator -- -j 4
cmake-build-debug/scientific_calculator
```

#### **Scientific Calculator License**

Copyright © 2019 by Intelligent Embedded Systems, University of Kassel, Germany

Permission to use, copy, modify, and distribute this software and its documentation under the terms of the GNU General Public License is hereby granted. No representations are made about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty. See the GNU General Public License for more details.

2 Main Page

# **Chapter 2**

# **Data Structure Index**

# 2.1 Data Structures

Here are the data structures with brief descriptions:

Complex		
Maduly	Struct representing a complex number consisting of a real and an imaginary number	7
Matrix	Struct representing a two dimensional matrix	8

Data Structure Index

# **Chapter 3**

# File Index

# 3.1 File List

Here is a list of all documented files with brief descriptions:

calculator.c								
Calculator for scientific computing	 	 					 	ç
math_library.c								
Mathematical functions for scientific computing	 	 						10
math_library.h								
Mathematical functions for scientific computing	 	 						18

6 File Index

# Chapter 4

# **Data Structure Documentation**

# 4.1 Complex Struct Reference

Struct representing a complex number consisting of a real and an imaginary number.

```
#include <math_library.h>
```

#### **Data Fields**

- float real
- float imag

# 4.1.1 Detailed Description

Struct representing a complex number consisting of a real and an imaginary number.

See also

```
https://en.wikipedia.org/wiki/Complex_number (last access: 23.05.2019)
```

#### 4.1.2 Field Documentation

#### 4.1.2.1 imag

Complex::imag

Member 'imag' represents the imaginary valued part of the complex number.

4.1.2.2 real

Complex::real

Member 'real' represents the real valued part of the complex number.

The documentation for this struct was generated from the following file:

• math\_library.h

# 4.2 Matrix Struct Reference

Struct representing a two dimensional matrix.

```
#include <math_library.h>
```

#### **Data Fields**

- int n\_rows
- int n\_columns
- float \*\* values

# 4.2.1 Detailed Description

Struct representing a two dimensional matrix.

#### 4.2.2 Field Documentation

#### 4.2.2.1 n\_columns

```
Matrix::n_columns
```

Member 'n\_columns' represents the number of columns of the matrix.

```
4.2.2.2 n_rows
```

```
Matrix::n_rows
```

Member 'n\_rows' represents the number of rows of the matrix.

#### 4.2.2.3 values

Matrix::values

Member 'values' is a double pointer containing the values of the matrix.

#### See also

```
https://en.wikipedia.org/wiki/Matrix_(mathematics) (last access: 23.05.2019)
```

The documentation for this struct was generated from the following file:

• math\_library.h

# **Chapter 5**

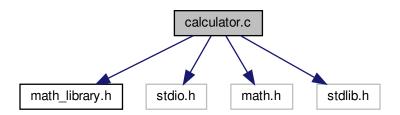
# **File Documentation**

# 5.1 calculator.c File Reference

Calculator for scientific computing.

```
#include "math_library.h"
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
```

Include dependency graph for calculator.c:



# **Functions**

• int main ()

Main function for execution of scientific calculator.

# 5.1.1 Detailed Description

Calculator for scientific computing.

Author

Marek Herde

Date

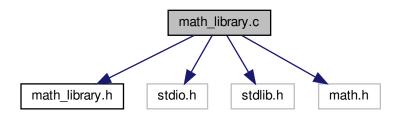
23.05.2019

# 5.2 math\_library.c File Reference

Mathematical functions for scientific computing.

```
#include "math_library.h"
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
```

Include dependency graph for math\_library.c:



#### **Functions**

• struct Matrix create\_matrix (int n\_rows, int n\_columns)

Allocates and creates a matrix with given input dimensions.

• struct Matrix read matrix ()

Reads in a matrix entered by the user.

• void print\_matrix (struct Matrix matrix)

Prints input matrix.

• struct Matrix transpose\_matrix (struct Matrix m)

This functions transposes a given input matrix.

• struct Matrix add\_matrices (struct Matrix m\_1, struct Matrix m\_2)

This functions adds two given input matrices.

• struct Matrix multiply\_matrices (struct Matrix m\_1, struct Matrix m\_2)

This functions computes the product of two given input matrices.

• float poly\_discriminant (float a, float b, float c)

Computes the discriminant of a polynomial function of degree two.

• int factorial (int n)

Computes the factorial of a positive integer.

# 5.2.1 Detailed Description

Mathematical functions for scientific computing.

**Author** 

Marek Herde

Date

23.05.2019

This library contains several mathematical functions and a list of them is given below.

- · Transpose matrix.
- · Add two matrices.
- · Multiply two matrices.
- · Compute roots of quadratic polynomial.
- · Compute factorial.

#### 5.2.2 Function Documentation

#### 5.2.2.1 add\_matrices()

This functions adds two given input matrices.

Denoting the matrix by resulting matrix by M and the input matrices by X and Y, the addition is described by  $M = X \cdot Y$ .

#### **Parameters**

<i>m</i> ⊷ _1	first matrix
m⊷	second matrix
_2	

#### Returns

sum of the matrices 'm\_1' and 'm\_2'

Here is the call graph for this function:



Here is the caller graph for this function:



# 5.2.2.2 create\_matrix()

Allocates and creates a matrix with given input dimensions.

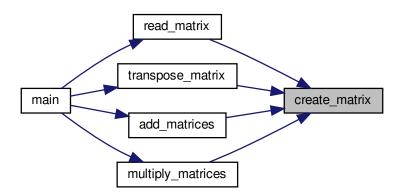
#### **Parameters**

n_rows	number of rows				
n_columns	number of columns				

# Returns

created matrix

Here is the caller graph for this function:



#### 5.2.2.3 factorial()

```
int factorial ( \quad \text{ int } n \ )
```

Computes the factorial of a positive integer.

#### **Parameters**

```
n being a positive integer whose factorial is to be computed
```

#### Returns

factorial of the parameter

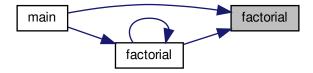
#### See also

```
https://en.wikipedia.org/wiki/Factorial (last access: 23.05.2019)
```

Here is the call graph for this function:



Here is the caller graph for this function:



# 5.2.2.4 multiply\_matrices()

This functions computes the product of two given input matrices.

Denoting the product matrix by  $M \in \mathbb{R}^{n \times m}$  and the input matrices by  $X \in \mathbb{R}^{n \times t}$  and  $Y \in \mathbb{R}^{t \times m}$ , the product is described by  $M = X \cdot Y$ .

#### **Parameters**

m⊷	first matrix
_1	
m⊷	second matrix
_2	

#### Returns

product of the matrices 'm\_1' and 'm\_2'

Here is the call graph for this function:



Here is the caller graph for this function:



# 5.2.2.5 poly\_discriminant()

```
float poly_discriminant (  \label{float a, float b, float b, float c}
```

Computes the discriminant of a polynomial function of degree two.

A polynomial function of degree two is defined by  $f(x)=ax^2+bx+c$ . The corresponding discriminant is given by  $b^2-4ac$ .

# **Parameters**

а	coefficient of polynomial of second order
b	coefficient of polynomial of first order
Gene	coefficient of polynomial of zeroth order

#### Returns

discriminant of polynomial function

#### See also

```
https://en.wikipedia.org/wiki/Discriminant
```

Here is the caller graph for this function:



# 5.2.2.6 print\_matrix()

Prints input matrix.

#### **Parameters**

m matrix to be printed

Here is the caller graph for this function:



# 5.2.2.7 read\_matrix()

```
struct Matrix read_matrix ( )
```

Reads in a matrix entered by the user.

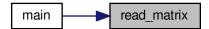
Returns

entered matrix

Here is the call graph for this function:



Here is the caller graph for this function:



# 5.2.2.8 transpose\_matrix()

```
struct Matrix transpose_matrix (
    struct Matrix m )
```

This functions transposes a given input matrix.

Denoting the input matrix by M, the transpose  $M^{T}$  is returned.

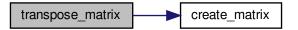
#### **Parameters**

m matrix to be transposed

Returns

transposed matrix

Here is the call graph for this function:



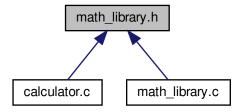
Here is the caller graph for this function:



# 5.3 math\_library.h File Reference

Mathematical functions for scientific computing.

This graph shows which files directly or indirectly include this file:



# **Data Structures**

struct Complex

Struct representing a complex number consisting of a real and an imaginary number.

struct Matrix

Struct representing a two dimensional matrix.

#### **Functions**

• struct Matrix create\_matrix (int n\_rows, int n\_columns)

Allocates and creates a matrix with given input dimensions.

struct Matrix read\_matrix ()

Reads in a matrix entered by the user.

void print matrix (struct Matrix m)

Prints input matrix.

struct Matrix transpose\_matrix (struct Matrix m)

This functions transposes a given input matrix.

struct Matrix add\_matrices (struct Matrix m\_1, struct Matrix m\_2)

This functions adds two given input matrices.

• struct Matrix multiply\_matrices (struct Matrix m\_1, struct Matrix m\_2)

This functions computes the product of two given input matrices.

float poly\_discriminant (float a, float b, float c)

Computes the discriminant of a polynomial function of degree two.

• int factorial (int n)

Computes the factorial of a positive integer.

float sine (float angle)

Use brief, otherwise the index won't have a brief explanation.

#### 5.3.1 Detailed Description

Mathematical functions for scientific computing.

**Author** 

Marek Herde

Date

23.05.2019

This library contains several mathematical functions and a list of them is given below.

- · Transpose matrix.
- · Add two matrices.
- · Multiply two matrices.
- · Compute roots of quadratic polynomial.
- · Compute factorial.

#### 5.3.2 Function Documentation

#### 5.3.2.1 add\_matrices()

This functions adds two given input matrices.

Denoting the matrix by resulting matrix by M and the input matrices by X and Y, the addition is described by  $M = X \cdot Y$ .

# **Parameters**

m⊷	first matrix
_1	
m⊷	second matrix
_2	

# Returns

sum of the matrices 'm\_1' and 'm\_2'

Here is the call graph for this function:



Here is the caller graph for this function:



# 5.3.2.2 create\_matrix()

Allocates and creates a matrix with given input dimensions.

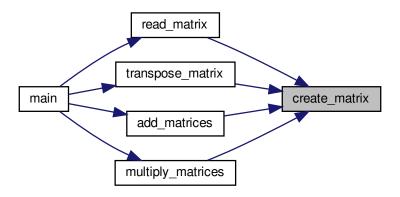
# Parameters

n_rows	number of rows					
n columns	number of columns					

#### Returns

created matrix

Here is the caller graph for this function:



# 5.3.2.3 factorial()

```
int factorial ( \quad \text{int } n \ )
```

Computes the factorial of a positive integer.

#### **Parameters**

n being a positive integer whose factorial is to be computed

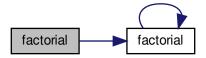
# Returns

factorial of the parameter

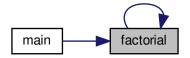
See also

```
https://en.wikipedia.org/wiki/Factorial (last access: 23.05.2019)
```

Here is the call graph for this function:



Here is the caller graph for this function:



# 5.3.2.4 multiply\_matrices()

This functions computes the product of two given input matrices.

Denoting the product matrix by  $M \in \mathbb{R}^{n \times m}$  and the input matrices by  $X \in \mathbb{R}^{n \times t}$  and  $Y \in \mathbb{R}^{t \times m}$ , the product is described by  $M = X \cdot Y$ .

#### **Parameters**

<i>m</i> ⊷ _1	first matrix
m←	second matrix
_2	

#### Returns

product of the matrices 'm\_1' and 'm\_2'

Here is the call graph for this function:



Here is the caller graph for this function:



# 5.3.2.5 poly\_discriminant()

Computes the discriminant of a polynomial function of degree two.

A polynomial function of degree two is defined by  $f(x)=ax^2+bx+c$ . The corresponding discriminant is given by  $b^2-4ac$ .

#### **Parameters**

	coefficient of polynomial of second order
b	coefficient of polynomial of first order
С	coefficient of polynomial of zeroth order

#### Returns

discriminant of polynomial function

#### See also

https://en.wikipedia.org/wiki/Discriminant

Here is the caller graph for this function:



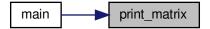
# 5.3.2.6 print\_matrix()

Prints input matrix.

#### **Parameters**

m matrix to be printed

Here is the caller graph for this function:



#### 5.3.2.7 read\_matrix()

```
struct Matrix read_matrix ( )
```

Reads in a matrix entered by the user.

Returns

entered matrix

Here is the call graph for this function:



Here is the caller graph for this function:



# 5.3.2.8 sine()

```
float sine (
          float angle )
```

Use brief, otherwise the index won't have a brief explanation.

Detailed explanation.

#### 5.3.2.9 transpose\_matrix()

This functions transposes a given input matrix.

Denoting the input matrix by M, the transpose  $M^T$  is returned.

#### **Parameters**

m matrix to be transposed

Returns

transposed matrix

Here is the call graph for this function:



Here is the caller graph for this function:

