Scientific Calculator

Generated by Doxygen 1.8.16

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

Chapter 2

Data Structure Index

2.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 3

File Index

3.1 File List

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

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

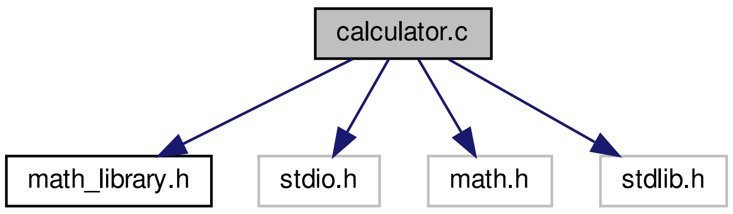
File Documentation

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

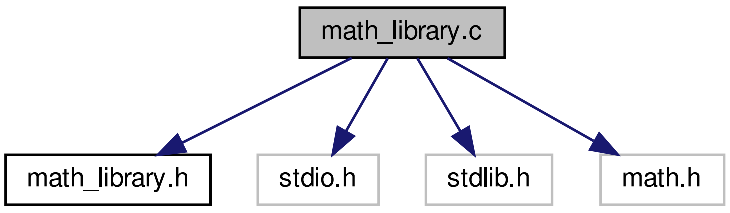
23.05.2019

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()

```
struct Matrix add_matrices (
            struct Matrix m_1,
            struct Matrix m_2 )
```

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 $M = X \cdot Y$.

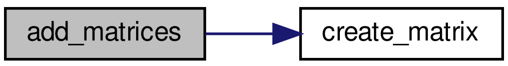
Parameters

m↩_1	first matrix
m↩_2	second matrix

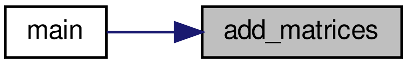
Returns

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()

```
struct Matrix create_matrix (
              int n_rows,
              int n_columns )
```

Allocates and creates a matrix with given input dimensions.

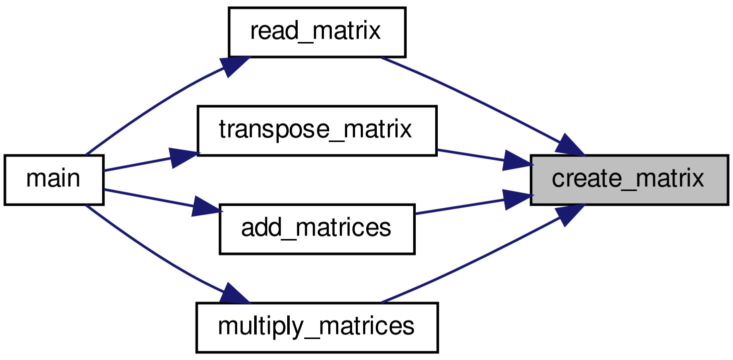
Parameters

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 (
            int n )
```

Computes the factorial of a positive integer.

Parameters

n	being a positive integer whose factorial is to be computed

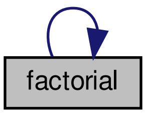
Returns

factorial of the parameter

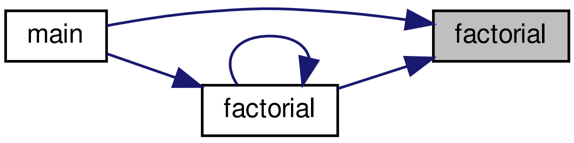
See also

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()

```
struct Matrix multiply_matrices (
            struct Matrix m_1,
            struct Matrix m_2 )
```

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 $M = X \cdot Y$.

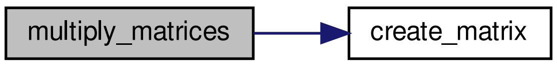
Parameters

m↩ _1	first matrix
m↩ _2	second matrix

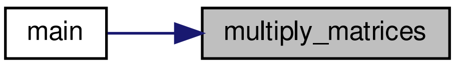
Returns

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 (
            float a,
            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

a	coefficient of polynomial of second order
b	coefficient of polynomial of first order
c	coefficient of polynomial of zeroth order

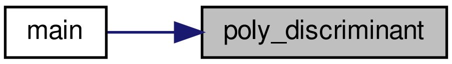
Returns

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()

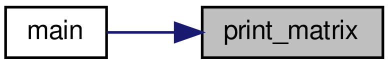
```
void print_matrix (
             struct Matrix m )
```

Prints input matrix.

Parameters

m	matrix to be printed

Here is the caller graph for this function:



5.2.2.7 read_matrix()

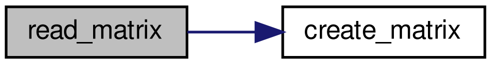
```
struct Matrix read_matrix ( )
```

Reads in a matrix entered by the user.

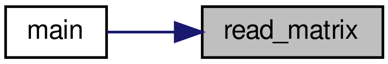
Returns

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.

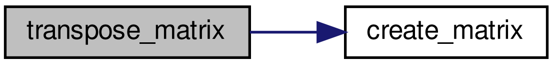
Parameters

m	matrix to be transposed

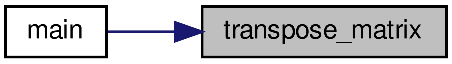
Returns

transposed matrix

Here is the call graph for this function:



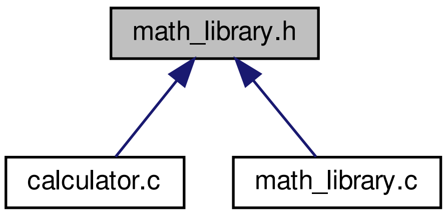
Here is the caller graph for this function:



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()

```
struct Matrix add_matrices (
            struct Matrix m_1,
            struct Matrix m_2 )
```

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 $M = X \cdot Y$.

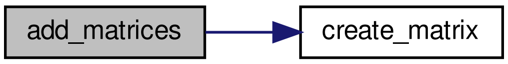
Parameters

m←↩ _1	first matrix
m←↩ _2	second matrix

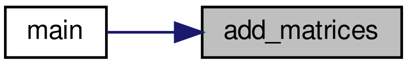
Returns

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()

```
struct Matrix create_matrix (
             int n_rows,
             int n_columns )
```

Allocates and creates a matrix with given input dimensions.

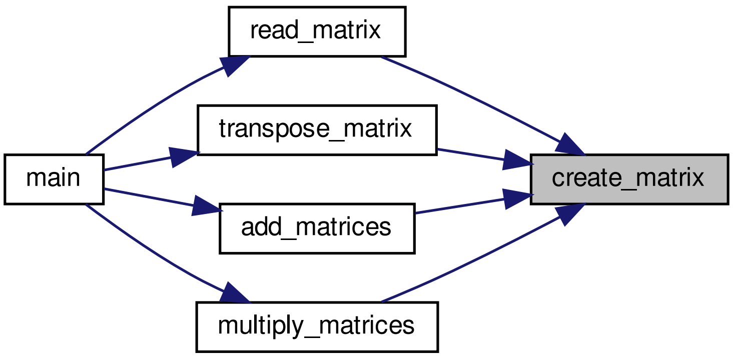
Parameters

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 (
            int n )
```

Computes the factorial of a positive integer.

Parameters

n	being a positive integer whose factorial is to be computed

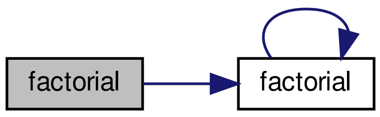
Returns

factorial of the parameter

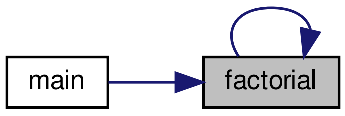
See also

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()

```
struct Matrix multiply_matrices (
              struct Matrix m_1,
              struct Matrix m_2 )
```

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 $M = X \cdot Y$.

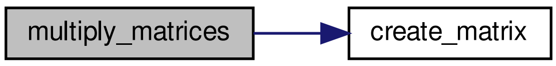
Parameters

m← *_1*	first matrix
m← *_2*	second matrix

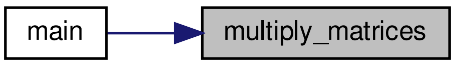
Returns

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()

```
float poly_discriminant (
            float a,
            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

a	coefficient of polynomial of second order
b	coefficient of polynomial of first order
c	coefficient of polynomial of zeroth order

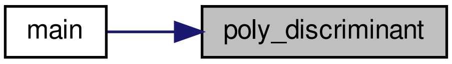
Returns

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()

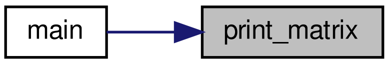
```
void print_matrix (
            struct Matrix m )
```

Prints input matrix.

Parameters

m	matrix to be printed

Here is the caller graph for this function:



5.3.2.7 read_matrix()

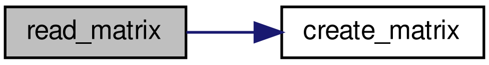
```
struct Matrix read_matrix ( )
```

Reads in a matrix entered by the user.

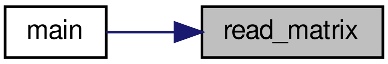
Returns

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()

```
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.

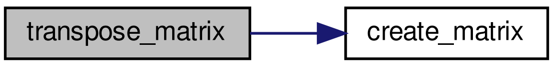
Parameters

m	matrix to be transposed

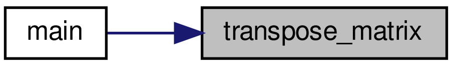
Returns

transposed matrix

Here is the call graph for this function:



Here is the caller graph for this function:


main
transpose_matrix