Quadratic Equation Solver

Generated by Doxygen 1.9.1

1 File Index	1
1.1 File List	1
2 File Documentation	3
2.1 main.cpp File Reference	3
2.1.1 Function Documentation	3
2.1.1.1 main()	3
2.2 quadr_solver.cpp File Reference	3
2.2.1 Function Documentation	4
2.2.1.1 are_doubles_equal()	4
2.2.1.2 ask_for_coeff()	5
2.2.1.3 clear_buffer()	5
2.2.1.4 get_coeffs()	5
2.2.1.5 getnum()	5
2.2.1.6 max_of_two()	6
2.2.1.7 min_of_two()	6
2.2.1.8 print_roots()	6
2.2.1.9 solve_lin()	7
2.2.1.10 solve_quadr()	7
2.3 quadr_solver.h File Reference	8
2.3.1 Macro Definition Documentation	9
2.3.1.1 MY_ASSERT	9
2.3.2 Enumeration Type Documentation	9
2.3.2.1 roots_amount	9
2.3.3 Function Documentation	9
2.3.3.1 are_doubles_equal()	10
2.3.3.2 ask_for_coeff()	10
2.3.3.3 clear_buffer()	10
2.3.3.4 get_coeffs()	10
2.3.3.5 getnum()	11
2.3.3.6 max_of_two()	11
2.3.3.7 min_of_two()	11
2.3.3.8 print_roots()	12
2.3.3.9 solve_lin()	12
2.3.3.10 solve_quadr()	13
2.3.3.11 start_tests()	13
2.3.3.12 test_eq()	13
2.3.4 Variable Documentation	14
2.3.4.1 EPSILON	14
2.4 quadr_tests.cpp File Reference	14
2.4.1 Function Documentation	14
2.4.1.1 start_tests()	14

2.4.1.2 test_eq()		
Index	14	2.4.1.2 test_eq()
	17	dex

Chapter 1

File Index

1.1 File List

Here is a list of all files with brief descriptions:

main.cpp		 										 									(3
quadr_solver.cpp)	 										 									;	3
quadr_solver.h		 										 									1	5
quadr tests.cpp		 										 									14	4

2 File Index

Chapter 2

File Documentation

2.1 main.cpp File Reference

```
#include "quadr_solver.h"
```

Functions

• int main (int argc, char *argv[])

2.1.1 Function Documentation

2.1.1.1 main()

```
int main (
          int argc,
          char * argv[] )
```

2.2 quadr_solver.cpp File Reference

```
#include "quadr_solver.h"
```

Functions

• void **getnum** (double *num)

Reads input data. If the data is not a valid number prints "This does not seem to be a valid number" and tries to read again.

• void **print_roots** (int **roots_amount**, double *x1, double *x2)

Prints the solutions of the equation if they exits. Prints "x can be any real number." if there are infinite solutions and "No real solutions." if there are no real solutions.

• int solve_quadr (double a, double b, double c, double *x1, double *x2)

```
Solves equation ax^2 + bx + c = 0.
```

• int solve_lin (double a, double b, double *x)

Solves equation ax + b = 0.

• int are_doubles_equal (double n1, double n2)

Returns 1 if $|n1 - n2| \le EPSILON$ and 0 otherwise.

void clear_buffer (void)

Clears input buffer using getchar() function.

void ask_for_coeff (char coeff_name, double *coeff_address)

Prints a prompt and reads input data using getnum() (p. 11) function.

void get_coeffs (double *a, double *b, double *c)

Gets coefficients of the equation.

double max_of_two (double n1, double n2)

Identifies which of two numbers is larger.

• double min_of_two (double n1, double n2)

Identifies which of two numbers is smaller.

2.2.1 Function Documentation

2.2.1.1 are_doubles_equal()

```
int are_doubles_equal ( \label{eq:double_n1} \mbox{double } n1, \\ \mbox{double } n2 \mbox{ )}
```

Returns 1 if $|n1 - n2| \le EPSILON$ and 0 otherwise.

Parameters

in	n1	First number.
in	n2	Second number.

Returns

Returns 1 or 0.

2.2.1.2 ask_for_coeff()

Prints a prompt and reads input data using **getnum()** (p. 11) function.

Parameters

in	coeff_name	Name of the coefficient.
in	coeff_address	Pointer to the coefficient.

2.2.1.3 clear_buffer()

```
void clear_buffer (
     void )
```

Clears input buffer using getchar() function.

2.2.1.4 get_coeffs()

```
void get_coeffs ( \label{eq:coeffs} \mbox{double * a,} \\ \mbox{double * b,} \\ \mbox{double * c )}
```

Gets coefficients of the equation.

Parameters

out	а	Pointer to a-coefficient of the equation.
out	b	Pointer to b-coefficient of the equation.
out	С	Pointer to c-coefficient of the equation.

2.2.1.5 getnum()

Reads input data. If the data is not a valid number prints "This does not seem to be a valid number" and tries to read again.

Parameters

in <i>num</i> Pointer to the number.	in	num	Pointer to the number.
------------------------------------------	----	-----	------------------------

2.2.1.6 max_of_two()

Identifies which of two numbers is larger.

Parameters

in	n1	First number.
in	n2	Second number.

Returns

Returns n1 if n1 > n2 and n2 otherwise.

2.2.1.7 min_of_two()

Identifies which of two numbers is smaller.

Parameters

in	n1	First number.
in	n2	Second number.

Returns

Returns n2 if n1 > n2 and n2 otherwise.

2.2.1.8 print_roots()

```
void print_roots (
          int roots_amount,
```

```
double * x1, double * x2)
```

Prints the solutions of the equation if they exits. Prints "x can be any real number." if there are infinite solutions and "No real solutions." if there are no real solutions.

Parameters

in	roots_amount	Amount of the solutions of the equation.
in	x1	Pointer to x1.
in	x2	Pointer to x2.

2.2.1.9 solve_lin()

```
int solve_lin (  \label{eq:condition} \mbox{double $a$,} \\ \mbox{double $b$,} \\ \mbox{double $*$ $x$ )}
```

Solves equation ax + b = 0.

Parameters

in	а	a-coefficient of the equation.	
in	b	b-coefficient of the equation.	
out	Х	Pointer to the solution of the equation.	

Returns

Returns the amount of the solutions of the equation and INF_ROOTS if the equation has infinite solutions.

2.2.1.10 solve_quadr()

Solves equation $ax^2 + bx + c = 0$.

Parameters

in	a a-coefficient of the equation.		
in	in b b-coefficient of the equation.		
in	С	c-coefficient of the equation.	
Generated by Doxyge Pointer to one of the solutions of the equation			
out	x2	Pointer to one of the solutions of the equation.	

Returns

Returns the amount of the solutions of the equation and INF_ROOTS if the equation has infinite solutions.

2.3 quadr solver.h File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <assert.h>
#include <string.h>
```

Macros

#define MY_ASSERT(condition) if (!(condition)) printf("Error %s in line %d in function %s\n", #condition,
 __LINE___, __func__) \

If condition is false, prints an error message. Does not abort the program.

Enumerations

• enum roots_amount { NO_ROOTS, ONE_ROOT, TWO_ROOTS, INF_ROOTS = -1 } Possible amounts of solutions of the equation.

Functions

void getnum (double *num)

Reads input data. If the data is not a valid number prints "This does not seem to be a valid number" and tries to read again.

void print_roots (int roots_amount, double *x1, double *x2)

Prints the solutions of the equation if they exits. Prints "x can be any real number." if there are infinite solutions and "No real solutions." if there are no real solutions.

• int solve_quadr (double a, double b, double c, double *x1, double *x2)

Solves equation $ax^2 + bx + c = 0$.

int solve_lin (double a, double b, double *x)

Solves equation ax + b = 0.

• int are_doubles_equal (double n1, double n2)

Returns 1 if $|n1 - n2| \le EPSILON$ and 0 otherwise.

void clear buffer (void)

Clears input buffer using getchar() function.

• void ask for coeff (char coeff name, double *coeff address)

Prints a prompt and reads input data using getnum() (p. 11) function.

void get_coeffs (double *a, double *b, double *c)

Gets coefficients of the equation.

• double max_of_two (double n1, double n2)

Identifies which of two numbers is larger.

double min_of_two (double n1, double n2)

Identifies which of two numbers is smaller.

- void **test_eq** (double a, double b, double c, int expected_nRoots, double expected_x1, double expected_x2)

 Tests if **solve_quadr()** (p. 12) works as expected.
- void start_tests (void)

Launches the tests. Gets test data from test_eq_data.txt file. Prints an error message if the file was not found.

Variables

const double EPSILON = 1e-7
 Determines precision of are_doubles_equal() (p. 9).

2.3.1 Macro Definition Documentation

2.3.1.1 MY_ASSERT

If condition is false, prints an error message. Does *not* abort the program.

Parameters

in <i>condition</i> Condition	١.	
-------------------------------	----	--

2.3.2 Enumeration Type Documentation

2.3.2.1 roots_amount

```
enum roots_amount
```

Possible amounts of solutions of the equation.

Enumerator

NO_ROOTS	
ONE_ROOT	
TWO_ROOTS	
INF_ROOTS	

2.3.3 Function Documentation

2.3.3.1 are_doubles_equal()

```
int are_doubles_equal ( \label{eq:double_n1} \mbox{double } n1, \\ \mbox{double } n2 \mbox{ )}
```

Returns 1 if $|n1 - n2| \le EPSILON$ and 0 otherwise.

Parameters

in	n1	First number.
in	n2	Second number.

Returns

Returns 1 or 0.

2.3.3.2 ask_for_coeff()

Prints a prompt and reads input data using **getnum()** (p. 11) function.

Parameters

in	coeff_name	Name of the coefficient.
in	coeff_address	Pointer to the coefficient.

2.3.3.3 clear_buffer()

```
void clear_buffer (
     void )
```

Clears input buffer using getchar() function.

2.3.3.4 get_coeffs()

Gets coefficients of the equation.

Parameters

out	a Pointer to a-coefficient of the equa	
out	b Pointer to b-coefficient of the equation	
out	С	Pointer to c-coefficient of the equation.

2.3.3.5 getnum()

```
void getnum ( \label{eq:condition} \mbox{double * num )}
```

Reads input data. If the data is not a valid number prints "This does not seem to be a valid number" and tries to read again.

Parameters

in <i>num</i>	Pointer to the number.
---------------	------------------------

2.3.3.6 max_of_two()

Identifies which of two numbers is larger.

Parameters

in	n1	First number.
in	n2	Second number.

Returns

Returns n1 if n1 > n2 and n2 otherwise.

2.3.3.7 min_of_two()

```
double min_of_two ( \label{eq:condition} \mbox{double } n1, \\ \mbox{double } n2 \mbox{ )}
```

Identifies which of two numbers is smaller.

Parameters

in	n1	First number.
in	n2	Second number.

Returns

Returns n2 if n1 > n2 and n2 otherwise.

2.3.3.8 print_roots()

```
void print_roots (
          int roots_amount,
           double * x1,
           double * x2 )
```

Prints the solutions of the equation if they exits. Prints "x can be any real number." if there are infinite solutions and "No real solutions." if there are no real solutions.

Parameters

in	roots_amount	Amount of the solutions of the equation.
in	x1	Pointer to x1.
in	x2	Pointer to x2.

2.3.3.9 solve_lin()

```
int solve_lin ( \label{eq:constraint} \mbox{double $a$,} \\ \mbox{double $b$,} \\ \mbox{double * $x$ )}
```

Solves equation ax + b = 0.

Parameters

in	а	a-coefficient of the equation.	
in	b	b-coefficient of the equation.	
out	Х	Pointer to the solution of the equation.	

Returns

Returns the amount of the solutions of the equation and INF_ROOTS if the equation has infinite solutions.

2.3.3.10 solve_quadr()

Solves equation $ax^2 + bx + c = 0$.

Parameters

in	а	a-coefficient of the equation.
in	b	b-coefficient of the equation.
in	С	c-coefficient of the equation.
out	x1	Pointer to one of the solutions of the equation.
out	x2	Pointer to one of the solutions of the equation.

Returns

Returns the amount of the solutions of the equation and INF_ROOTS if the equation has infinite solutions.

2.3.3.11 start_tests()

```
void start_tests (
     void )
```

Launches the tests. Gets test data from test_eq_data.txt file. Prints an error message if the file was not found.

2.3.3.12 test_eq()

Tests if **solve_quadr()** (p. 12) works as expected.

Parameters

in	а	a-coefficient of the equation.
in	b	b-coefficient of the equation.
in	С	c-coefficient of the equation.
in	expected_nRoots	Expected amount of the solutions of the equation.
in	by Doxygen expected_x1	Expected x1.
in	expected_x2	Expected x2.

2.3.4 Variable Documentation

2.3.4.1 EPSILON

```
const double EPSILON = 1e-7
```

Determines precision of are_doubles_equal() (p. 9).

2.4 quadr_tests.cpp File Reference

```
#include "quadr_solver.h"
```

Functions

- void **test_eq** (double a, double b, double c, int expected_nRoots, double expected_x1, double expected_x2)

 Tests if **solve_quadr()** (p. 12) works as expected.
- void start_tests (void)

Launches the tests. Gets test data from test_eq_data.txt file. Prints an error message if the file was not found.

2.4.1 Function Documentation

2.4.1.1 start_tests()

```
void start_tests (
     void )
```

Launches the tests. Gets test data from test_eq_data.txt file. Prints an error message if the file was not found.

2.4.1.2 test_eq()

Tests if solve_quadr() (p. 12) works as expected.

Parameters

in	а	a-coefficient of the equation.
in	Ь	b-coefficient of the equation.
in	С	c-coefficient of the equation.
in	expected_nRoots	Expected amount of the solutions of the equation.
in	expected_x1	Expected x1.
in	expected_x2	Expected x2.

Index

quadr_solver.h, 9 ask_for_coeff quadr_solver.cpp, 4 quadr_solver.cpp, 5 quadr_solver.cpp, 5 quadr_solver.cpp, 5 quadr_solver.h, 10 EPSILON quadr_solver.cpp, 5 quadr_solver.cpp, 5 quadr_solver.cpp, 5 quadr_solver.cpp, 5 quadr_solver.h, 14 get_coeffs quadr_solver.cpp, 5 quadr_solver.h, 10 get_coeffs quadr_solver.cpp, 5 quadr_solver.h, 10 get_main quadr_solver.cpp, 5 quadr_solver.h, 11 INF_ROOTS, 9 INF_ROOTS quadr_solver.h, 11 INF_ROOTS quadr_solver.h, 9 main.cpp, 3 main.cpp, 3 main.cpp, 3 main.cpp, 3 main, 3 main.cpp, 3 max_of_two quadr_solver.cpp, 6 quadr_solver.h, 11 MY_ASSERT quadr_solver.h, 11 MY_ASSERT quadr_solver.h, 11 MY_ASSERT quadr_solver.pp, 6 quadr_solver.pp, 6 quadr_solver.pp, 6 quadr_solver.h, 11 MY_ASSERT quadr_solver.h, 9 NO_ROOTS quadr_solver.pp, 6 quadr_solver.pp, 6 quadr_solver.h, 9 NO_ROOTS quadr_solver.pp, 7 quadr_solver.pp, 14 test_eq quadr_solver.pp, 14 test_eq quadr_solver.pp, 14 quadr_solver.pp, 14 quadr_solver.pp, 14	are_doubles_equal	get_coeffs, 5
ask_for_coeff	quadr_solver.cpp, 4	getnum, 5
quadr_solver.cpp, 4 quadr_solver.cp, 5 quadr_solver.cpp, 5 quadr_solver.cpp, 5 quadr_solver.cpp, 5 quadr_solver.cp, 10 EPSILON quadr_solver.cp, 5 quadr_solver.cp, 14 get_coeffs quadr_solver.cpp, 5 quadr_solver.h, 11 INF_ROOTS	• —	max_of_two, 6
quadr_solver.h, 10 clear_buffer quadr_solver.cpp, 5 quadr_solver.h, 10 EPSILON quadr_solver.h, 14 get_coeffs quadr_solver.cpp, 5 quadr_solver.h, 14 get_coeffs quadr_solver.h, 10 getnum quadr_solver.h, 10 getnum quadr_solver.h, 10 getnum quadr_solver.cpp, 5 quadr_solver.h, 11 min_of_two, 11 main.cpp, 3 main.cpp, 3 main.cpp, 3 main.gp, 3 main.g	ask_for_coeff	min_of_two, 6
clear_buffer quadr_solver.cpp, 5 quadr_solver.h, 10 EPSILON quadr_solver.h, 14 get_coeffs, 10 get_coeffs quadr_solver.cpp, 5 quadr_solver.h, 14 get_coeffs, 10 get_coeffs quadr_solver.cpp, 5 quadr_solver.cpp, 5 quadr_solver.cpp, 5 quadr_solver.cpp, 5 quadr_solver.cpp, 5 quadr_solver.cpp, 5 quadr_solver.h, 11 INF_ROOTS, 9 max_of_two, 11 min_of_two, 11 main.cpp, 3 main.cpp, 3 main.qp, 3 main, 3 max_of_two quadr_solver.cpp, 6 quadr_solver.h, 11 min_of_two quadr_solver.cpp, 6 quadr_solver.h, 11 my_ASSERT quadr_solver.h, 11 min_of_two quadr_solver.cpp, 6 quadr_solver.h, 11 my_ASSERT quadr_solver.h, 9 NO_ROOTS quadr_solver.h, 12 start_tests, 14 test_eq, 14 test_eq, 14 solve_lin quadr_solver.h, 9 Toots_amount quadr_solver.cpp, 7 quadr_solver.cpp, 7 quadr_solver.h, 12 start_tests quadr_solver.h, 12 start_tests quadr_solver.h, 12 start_tests quadr_solver.h, 13 quadr_tests.cpp, 14 test_eq quadr_solver.h, 13 quadr_tests.cpp, 14 test_eq quadr_solver.h, 13 quadr_tests.cpp, 14	quadr_solver.cpp, 4	print_roots, 6
clear_buffer quadr_solver.cpp, 5 quadr_solver.h, 10 EPSILON quadr_solver.h, 14 get_coeffs quadr_solver.cpp, 5 quadr_solver.h, 11 INF_ROOTS, 9 max_of_two, 11 min_of_two, 11 min_of_two, 11 MY_ASSERT, 9 NO_ROOTS, 9 clear_buffer, 10 EPSILON, 14 get_coeffs, 10 getnum quadr_solver.cpp, 5 quadr_solver.cpp, 5 quadr_solver.h, 11 INF_ROOTS, 9 NO_ROOTS, 9 ONE_ROOT, 9 print_roots, 12 roots_amount, 9 solve_lin, 12 solve_quadr, 12 start_tests, 13 test_eq, 13 TWO_ROOTS, 9 quadr_solver.cpp, 6 quadr_solver.cpp, 6 quadr_solver.cpp, 6 quadr_solver.cpp, 6 quadr_solver.h, 11 min_of_two quadr_solver.cpp, 6 quadr_solver.h, 19 MY_ASSERT quadr_solver.h, 9 NO_ROOTS quadr_solver.h, 9 NO_ROOTS quadr_solver.h, 9 NO_ROOTS quadr_solver.h, 9 NO_ROOTS quadr_solver.h, 9 Solve_quadr quadr_solver.cpp, 7 quadr_solver.cpp, 7 quadr_solver.cpp, 7 quadr_solver.h, 12 start_tests quadr_solver.h, 12 start_tests quadr_solver.h, 13	quadr_solver.h, 10	solve_lin, 7
quadr_solver.cpp, 5 quadr_solver.h, 10 EPSILON quadr_solver.h, 14 get_coeffs quadr_solver.cpp, 5 quadr_solver.h, 10 getnum quadr_solver.cpp, 5 quadr_solver.h, 11 INF_ROOTS, 9 max_of_two, 11 min_of_two, 11 min_of_two, 11 ONE_ROOT, 9 print_roots, 12 roots_amount, 9 solve_lin, 12 solve_quadr, 12 start_tests, 13 test_eq, 13 TWO_ROOTS, 9 quadr_solver.h, 11 min_of_two quadr_solver.cpp, 6 quadr_solver.h, 11 MY_ASSERT quadr_solver.h, 11 MY_ASSERT quadr_solver.h, 9 NO_ROOTS quadr_solver.h, 12 solve_quadr quadr_solver.pp, 7 quadr_solver.pp, 7 quadr_solver.pp, 14 start_tests, 14 test_eq, 14 solve_lin quadr_solver.h, 9 NO_ROOTS quadr_solver.h, 12 solve_quadr quadr_solver.pp, 7 quadr_solver.pp, 7 quadr_solver.h, 12 solve_quadr quadr_solver.h, 12 solve_quadr quadr_solver.h, 13		solve_quadr, 7
quadr_solver.h, 10 EPSILON quadr_solver.h, 14 get_coeffs quadr_solver.cpp, 5 quadr_solver.cpp, 5 quadr_solver.cpp, 5 quadr_solver.cpp, 5 quadr_solver.h, 11 INF_ROOTS, 9 max_of_two, 11 min_of_two, 11 main.cpp, 3 max_of_two, 12 main.cpp, 3 max_of_two, 11 main.cpp, 3 max_of_two, 11 main.cpp, 3 max_of_two, 11 ONE_ROOTS, 9 ONE_ROOT, 9 print_roots, 12 roots_amount, 9 solve_quadr, 12 start_tests, 13 test_eq, 13 TWO_ROOTS, 9 quadr_solver.cpp, 6 quadr_solver.cpp, 6 quadr_solver.h, 11 MY_ASSERT quadr_solver.h, 11 min_of_two quadr_solver.cpp, 6 quadr_solver.h, 11 MY_ASSERT quadr_solver.h, 9 NO_ROOTS PNO_ROOTS quadr_solver.cpp, 6 quadr_solver.h, 9 NO_ROOTS quadr_solver.h, 9 NO_ROOTS quadr_solver.h, 9 NO_ROOTS quadr_solver.h, 9 Solve_lin quadr_solver.h, 9 Toots_amount quadr_solver.h, 9 solve_lin quadr_solver.h, 9 Toots_amount quadr_solver.h, 9 solve_lin quadr_solver.pp, 7 quadr_solver.h, 12 solve_quadr quadr_solver.h, 12 solve_quadr quadr_solver.h, 12 solve_quadr quadr_solver.h, 12 solve_quadr quadr_solver.h, 13 quadr_tests.cpp, 14 test_eq quadr_solver.h, 13 quadr_solver.h, 13 quadr_tests.cpp, 14	clear_buffer	quadr_solver.h, 8
EPSILON quadr_solver.h, 14 quadr_solver.h, 14 get_coeffs quadr_solver.cpp, 5 quadr_solver.cpp, 5 quadr_solver.cpp, 5 quadr_solver.cpp, 5 quadr_solver.h, 11 INF_ROOTS quadr_solver.h, 9 main.cpp, 3 main.cpp, 3 main.cpp, 3 main.gp, 3	quadr_solver.cpp, 5	are_doubles_equal, 9
EPSILON quadr_solver.h, 14 get_coeffs, 10 get_coeffs quadr_solver.cpp, 5 quadr_solver.cpp, 5 quadr_solver.cpp, 5 quadr_solver.cpp, 5 quadr_solver.h, 11 INF_ROOTS, 9 max_of_two, 11 min_of_two, 11 min_of_two, 11 MY_ASSERT, 9 NO_ROOTS, 9 ONE_ROOT, 9 print_roots, 12 roots_amount, 9 solve_lin, 12 solve_quadr, 12 start_tests, 13 main.cpp, 3 main.cpp, 3 main.cpp, 3 main.cpp, 3 main.sp, 3	quadr_solver.h, 10	ask_for_coeff, 10
quadr_solver.h, 14 get_coeffs, 10 get_coeffs quadr_solver.cpp, 5 quadr_solver.cpp, 5 quadr_solver.cpp, 5 quadr_solver.cpp, 5 quadr_solver.cpp, 5 quadr_solver.h, 11 INF_ROOTS, 9 INF_ROOTS, 9 INF_ROOTS, 9 INF_ROOTS, 9 INF_ROOTS quadr_solver.h, 9 INF_ROOTS INF_ROOTS quadr_solver.h, 9 INF_ROOTS INF_ROOTS quadr_solver.h, 9 INF_ROOTS INF_ROOTS quadr_solver.h, 9 INF_ROOTS quadr_solver.h, 9 INF_ROOTS quadr_solver.cpp, 6 quadr_solver.cpp, 6 quadr_solver.cpp, 6 quadr_solver.cpp, 6 quadr_solver.h, 11 INF_ROOTS quadr_solver.cpp, 6 quadr_solver.cpp, 6 quadr_solver.cpp, 6 quadr_solver.cpp, 6 quadr_solver.cpp, 7 quadr_solver.h, 9 INF_ROOTS quadr_solver.cpp, 7 quadr_solver.cpp, 7 quadr_solver.h, 9 INF_ROOTS quadr_solver.cpp, 7 quadr_solver.h, 12 INF_ROOTS print_roots quadr_solver.cpp, 7 quadr_solver.h, 12 INF_ROOTS print_roots quadr_solver.h, 12 INF_ROOTS print_roots quadr_solver.h, 13 quadr_solver.h, 13 quadr_solver.h, 13 quadr_solver.h, 13 quadr_solver.h, 13 quadr_solver.h, 13 quadr_solver.cpp, 14		clear_buffer, 10
get_coeffs quadr_solver.cpp, 5 quadr_solver.cpp, 5 quadr_solver.cpp, 5 quadr_solver.cpp, 5 quadr_solver.h, 10 getnum quadr_solver.cpp, 5 quadr_solver.h, 11 INF_ROOTS quadr_solver.h, 11 INF_ROOTS quadr_solver.h, 9 INF_ROOTS quadr_solver.h, 9 main.cpp, 3 main.cpp, 3 main.cpp, 3 main, 3 Madr_solver.cpp, 6 quadr_solver.cpp, 6 quadr_solver.cpp, 6 quadr_solver.cpp, 6 quadr_solver.h, 11 INF_ROOTS quadr_solver.cpp, 6 quadr_solver.cpp, 7 quadr_solver.cpp, 12 start_tests quadr_solver.h, 12 start_tests quadr_solver.h, 13 quadr_solver.cpp, 14 test_eq quadr_solver.h, 13 quadr_solver.h, 13 quadr_solver.h, 13 quadr_solver.cpp, 14		EPSILON, 14
quadr_solver.cpp, 5 quadr_solver.cpp, 5 quadr_solver.cpp, 5 quadr_solver.cpp, 5 quadr_solver.h, 11 INF_ROOTS quadr_solver.h, 9 max_of_two, 11 min_of_two, 11 MY_ASSERT, 9 NO_ROOTS, 9 ONE_ROOT, 9 print_roots, 12 roots_amount, 9 solve_lin, 12 solve_lin, 12 main.cpp, 3 main.cpp, 3 main.cpp, 3 main.cpp, 3 main.cpp, 3 main, 3 main, 3 main, 3 main, 3 main, 6 quadr_solver.cpp, 6 quadr_solver.cpp, 6 quadr_solver.cpp, 6 quadr_solver.cpp, 6 quadr_solver.cpp, 6 quadr_solver.cpp, 6 quadr_solver.h, 11 MY_ASSERT quadr_solver.h, 9 NO_ROOTS Quadr_solver.h, 9 NO_ROOTS quadr_solver.h, 9 ONE_ROOT quadr_solver.h, 9 ONE_ROOT quadr_solver.h, 9 Solve_quadr quadr_solver.cpp, 7 quadr_solver.cpp, 7 quadr_solver.h, 12 start_tests quadr_solver.h, 13 quadr_solver.h, 13 quadr_solver.cpp, 14 test_eq quadr_solver.h, 13 quadr_solver.h, 13 quadr_solver.h, 13 quadr_solver.h, 13 quadr_solver.cpp, 14	quadr_solver.h, 14	get_coeffs, 10
quadr_solver.cpp, 5 quadr_solver.h, 10 getnum quadr_solver.cpp, 5 quadr_solver.cpp, 5 quadr_solver.h, 11 NF_ROOTS quadr_solver.h, 9 main.cpp, 3 main.cpp, 5 no No.RoOTS, 9 main.cpp, 14 main.cpp, 3 main.cpp, 3 main.cpp, 5 no No.RoOTS, 9 main.cpp, 6 quadr_solver.h, 12 quadr_solver.h, 12 main.cpp, 3 main.cp	not coeffe	getnum, 11
quadr_solver.h, 10 getnum	_	INF ROOTS, 9
quadr_solver.n, 10 getnum quadr_solver.cpp, 5 quadr_solver.h, 11 Quadr_solver.h, 11 Quadr_solver.h, 11 Quadr_solver.h, 9 QNE_ROOTS quadr_solver.h, 9 quadr_solver.h, 9 quadr_solver.h, 9 quadr_solver.h, 9 quadr_solver.pp, 3 quadr_solver.cpp, 3 quadr_solver.cpp, 6 quadr_solver.cpp, 6 quadr_solver.h, 11 quadr_solver.h, 12 quadr_solver.h, 9 quadr_solver.h, 9 quadr_solver.h, 9 quadr_solver.h, 9 quadr_solver.h, 12 quadr_solver.h, 9 quadr_solver.h, 12 quadr_solver.h, 13 quadr_solver.h, 13 quadr_solver.h, 13 quadr_solver.h, 13 quadr_solver.h, 13 quadr_solver.h, 13 quadr_solver.cpp, 3		
quadr_solver.cpp, 5 quadr_solver.h, 11 NO_ROOTS, 9 ONE_ROOT, 9 print_roots, 12 roots_amount, 9 solve_lin, 12 solve_quadr, 12 start_tests, 13 main.cpp, 3 main.cpp, 3 main, 3 main.cpp, 3 main, 3 main.cpp, 3 main, 3 main, 3 max_of_two quadr_solver.cpp, 6 quadr_solver.h, 11 min_of_two quadr_solver.cpp, 6 quadr_solver.h, 11 my_ASSERT quadr_solver.h, 9 NO_ROOTS quadr_solver.h, 9 Solve_quadr quadr_solver.cpp, 7 quadr_solver.cpp, 7 quadr_solver.h, 12 solve_quadr quadr_solver.h, 12 solve_quadr quadr_solver.h, 12 start_tests quadr_solver.h, 12 start_tests quadr_solver.h, 13 quadr_solver.h, 13 quadr_solver.h, 13 quadr_solver.h, 13 quadr_solver.h, 13 quadr_solver.h, 13 quadr_solver.cpp, 7	• —	
quadr_solver.cpp, 5 quadr_solver.h, 11 INF_ROOTS quadr_solver.h, 9 main main main.cpp, 3 main.cpp, 3 main, 3 main, 3 main, 3 max_of_two quadr_solver.h, 11 min_of_two quadr_solver.cpp, 6 quadr_solver.h, 11 my_ASSERT quadr_solver.h, 9 NO_ROOTS quadr_solver.h, 9 Solve_quadr quadr_solver.pp, 7 quadr_solver.h, 12 solve_quadr quadr_solver.h, 13 quadr_solver.h, 13 quadr_solver.h, 13 quadr_solver.h, 13 quadr_solver.h, 13 quadr_solver.cpp, 7		
quadr_solver.n, 11 ONE_ROOT, 9 print_roots, 12 roots_amount, 9 solve_lin, 12 solve_quadr, 12 start_tests, 13 test_eq, 13 TWO_ROOTS, 9 quadr_solver.cpp, 6 quadr_solver.cpp, 6 quadr_solver.cpp, 6 quadr_solver.h, 11 MY_ASSERT quadr_solver.h, 9 NO_ROOTS quadr_solver.h, 9 NO_ROOTS quadr_solver.h, 9 ONE_ROOT quadr_solver.h, 9 ONE_ROOT quadr_solver.h, 9 TWO_ROOTS quadr_solver.h, 9 Solve_lin quadr_solver.h, 9 Solve_lin quadr_solver.h, 12 solve_quadr quadr_solver.h, 12 solve_quadr quadr_solver.h, 12 solve_quadr quadr_solver.h, 12 start_tests quadr_solver.h, 13 quadr_solver.cpp, 14		
INF_ROOTS quadr_solver.h, 9 main main.cpp, 3 main.cpp, 3 main, 3 main, 3 max_of_two quadr_solver.h, 11 min_of_two quadr_solver.cpp, 6 quadr_solver.h, 11 my_ASSERT quadr_solver.h, 9 NO_ROOTS quadr_solver.h, 9 NO_ROOTS quadr_solver.h, 9 ONE_ROOT quadr_solver.h, 9 ONE_ROOT quadr_solver.h, 9 ONE_ROOT quadr_solver.h, 9 Toots_amount quadr_solver.h, 9 solve_lin quadr_solver.h, 9 solve_lin quadr_solver.h, 12 solve_quadr quadr_solver.h, 12 start_tests quadr_solver.h, 13 quadr_solver.cpp, 14	quadr_solver.h, 11	_
quadr_solver.h, 9 main quadr_solver.h, 9 main.cpp, 3 main.cpp, 3 main, 3 main, 3 max_of_two quadr_solver.cpp, 6 quadr_solver.h, 11 min_of_two quadr_solver.h, 11 my_ASSERT quadr_solver.h, 9 NO_ROOTS quadr_solver.h, 9 NO_ROOTS quadr_solver.h, 9 NO_ROOTS quadr_solver.h, 9 NO_ROOTS quadr_solver.h, 9 ONE_ROOT quadr_solver.h, 9 ONE_ROOT quadr_solver.h, 9 color amount quadr_solver.h, 12 color amount quadr_solver.h, 13 quadr_solver.h, 13 quadr_solver.h, 13 quadr_solver.h, 13 quadr_solver.h, 13 quadr_solver.h, 13 quadr_solver.h, 14	INE DOOTS	
main solve_lin, 12 main solve_quadr, 12 main.cpp, 3 start_tests, 13 main.cpp, 3 test_eq, 13 main, 3 TWO_ROOTS, 9 max_of_two quadr_solver.cpp, 6 quadr_solver.h, 11 test_eq, 14 min_of_two quadr_solver.cpp, 6 quadr_solver.h, 11 my_ASSERT quadr_solver.h, 9 NO_ROOTS quadr_solver.h, 9 NO_ROOTS quadr_solver.h, 9 NO_ROOTS quadr_solver.h, 9 NO_ROOTS quadr_solver.h, 9 Solve_lin quadr_solver.h, 9 solve_lin quadr_solver.h, 12 solve_lin quadr_solver.cpp, 7 quadr_solver.cpp, 7 quadr_solver.h, 12 solve_quadr quadr_solver.h, 12 solve_quadr quadr_solver.h, 13	-	• —
main solve_quadr, 12 main.cpp, 3 start_tests, 13 main.cpp, 3 test_eq, 13 main, 3 TWO_ROOTS, 9 quadr_tests.cpp, 14 quadr_solver.cpp, 6 quadr_solver.h, 11 test_eq, 14 min_of_two quadr_solver.cpp, 6 quadr_solver.h, 11 MY_ASSERT quadr_solver.h, 9 NO_ROOTS quadr_solver.h, 9 NO_ROOTS quadr_solver.h, 9 ONE_ROOT quadr_solver.h, 9 Solve_quadr quadr_solver.h, 12 quadr_solver.h, 9 start_tests quadr_solver.h, 12 quadr_solver.h, 13 quadr_solver.cpp, 6 quadr_solver.cpp, 6 quadr_solver.cpp, 7 quadr_solver.h, 12 quadr_solver.h, 13 quadr_solver.h, 13 quadr_solver.cpp, 3	quadr_solver.n, 9	
main.cpp, 3 main.cpp, 3 main.cpp, 3 main.cpp, 3 main, 3 main, 3 max_of_two quadr_solver.cpp, 6 quadr_solver.h, 11 min_of_two quadr_solver.h, 11 my_ASSERT quadr_solver.h, 9 NO_ROOTS quadr_solver.h, 9 NO_ROOTS quadr_solver.h, 9 NO_ROOTS quadr_solver.h, 9 NO_ROOTS quadr_solver.h, 9 Solve_quadr quadr_solver.h, 12 quadr_solver.h, 9 solve_quadr quadr_solver.h, 12 quadr_solver.h, 12 quadr_solver.h, 13 quadr_solver.cpp, 6 quadr_solver.cpp, 6 quadr_solver.cpp, 6 quadr_solver.h, 12 quadr_solver.h, 12 quadr_solver.cpp, 6 quadr_solver.cpp, 3 quadr_solver.h, 13 quadr_solver.cpp, 3	main	
main.cpp, 3 main, 3 max_of_two quadr_solver.cpp, 6 quadr_solver.h, 11 min_of_two quadr_solver.h, 11 my_ASSERT quadr_solver.h, 9 NO_ROOTS quadr_solver.h, 9 Tools_amount quadr_solver.h, 9 quadr_solver.h, 9 NO_ROOTS quadr_solver.h, 12 solve_quadr quadr_solver.cpp, 7 quadr_solver.cpp, 7 quadr_solver.h, 12 start_tests quadr_solver.h, 12 start_tests quadr_solver.h, 13 quadr_solver.cpp, 14 quadr_solver.cpp, 3 quadr_solver.h, 13 quadr_solver.h, 13 quadr_solver.h, 13 quadr_solver.cpp, 3		
main, 3 max_of_two quadr_solver.cpp, 6 quadr_solver.h, 11 min_of_two quadr_solver.h, 11 my_ASSERT quadr_solver.h, 9 NO_ROOTS quadr_solver.h, 9 NO_ROOTS quadr_solver.h, 9 NO_ROOTS quadr_solver.h, 9 NO_ROOTS quadr_solver.h, 9 Solve_quadr quadr_solver.h, 9 Solve_quadr quadr_solver.h, 12 solve_quadr quadr_solver.h, 12 solve_quadr quadr_solver.h, 12 start_tests quadr_solver.h, 13 quadr_solver.h, 12 test_eq quadr_solver.h, 13 quadr_solver.h, 13 quadr_solver.cpp, 3		
max_of_two quadr_tests.cpp, 14 quadr_solver.cpp, 6 quadr_solver.h, 11 min_of_two quadr_solver.cpp, 6 quadr_solver.h, 11 MY_ASSERT quadr_solver.h, 9 NO_ROOTS quadr_solver.h, 9 NOE_ROOT quadr_solver.h, 9 ONE_ROOT quadr_solver.h, 9 solve_quadr quadr_solver.h, 12 quadr_solver.h, 9 start_tests quadr_solver.h, 12 start_tests quadr_solver.h, 13 quadr_solver.cpp, 6 quadr_solver.h, 12 test_eq quadr_solver.h, 13	• •	
quadr_solver.cpp, 6 quadr_solver.h, 11 min_of_two quadr_solver.cpp, 6 quadr_solver.h, 11 MY_ASSERT quadr_solver.h, 9 NO_ROOTS quadr_solver.h, 9 NONE_ROOT quadr_solver.h, 9 ONE_ROOT quadr_solver.h, 9 start_tests, 14 test_eq, 14 roots_amount quadr_solver.h, 9 solve_lin quadr_solver.cpp, 7 quadr_solver.h, 12 solve_quadr quadr_solver.h, 12 solve_quadr quadr_solver.cpp, 7 quadr_solver.h, 12 quadr_solver.h, 13 quadr_solver.h, 13 quadr_solver.h, 12 test_eq quadr_solver.h, 13		
quadr_solver.h, 11 min_of_two quadr_solver.cpp, 6 quadr_solver.h, 11 MY_ASSERT quadr_solver.h, 9 NO_ROOTS quadr_solver.h, 9 NOE_ROOT quadr_solver.h, 9 ONE_ROOT quadr_solver.h, 9 solve_quadr quadr_solver.cpp, 7 quadr_solver.cpp, 7 quadr_solver.cpp, 7 quadr_solver.h, 12 start_tests quadr_solver.h, 13 quadr_solver.h, 12 quadr_solver.cpp, 6 quadr_solver.h, 12 quadr_solver.h, 12 quadr_solver.h, 12 quadr_solver.h, 12 quadr_solver.h, 13		. –
min_of_two quadr_solver.cpp, 6 quadr_solver.h, 11 MY_ASSERT quadr_solver.h, 9 NO_ROOTS quadr_solver.h, 9 ONE_ROOT quadr_solver.h, 9 ONE_ROOT quadr_solver.h, 9 solve_quadr quadr_solver.cpp, 7 quadr_solver.cpp, 7 quadr_solver.h, 12 solve_quadr quadr_solver.h, 12 start_tests quadr_solver.h, 13 quadr_solver.cpp, 6 quadr_solver.h, 12 test_eq quadr_solver.h, 13	. –	
quadr_solver.cpp, 6 quadr_solver.h, 11 MY_ASSERT quadr_solver.h, 9 NO_ROOTS quadr_solver.h, 9 ONE_ROOT quadr_solver.h, 9 ONE_ROOT quadr_solver.h, 9 one print_roots quadr_solver.cpp, 6 quadr_solver.h, 12 quadr_solver.cpp, 6 quadr_solver.h, 12 quadr_solver.h, 12 quadr_solver.h, 12 quadr_solver.h, 12 quadr_solver.h, 12 quadr_solver.h, 12 quadr_solver.h, 13	• —	tost_eq, 14
quadr_solver.h, 11 MY_ASSERT quadr_solver.h, 9 NO_ROOTS quadr_solver.h, 9 ONE_ROOT quadr_solver.h, 9 ONE_ROOT quadr_solver.h, 9 orint_roots quadr_solver.cpp, 6 quadr_solver.h, 12 quadr_solver.h, 12 quadr_solver.cpp, 6 quadr_solver.h, 12 quadr_solver.h, 12 quadr_solver.h, 12 quadr_solver.h, 12 quadr_solver.h, 13		roots amount
MY_ASSERT quadr_solver.h, 9 NO_ROOTS quadr_solver.h, 9 ONE_ROOT quadr_solver.h, 9 ONE_ROOT quadr_solver.h, 9 orint_roots quadr_solver.cpp, 6 quadr_solver.h, 12 quadr_solver.h, 12 quadr_solver.cpp, 6 quadr_solver.h, 12 quadr_solver.h, 12 quadr_solver.h, 12 quadr_solver.h, 13 quadr_solver.h, 13 quadr_solver.h, 12 quadr_solver.h, 13		
quadr_solver.h, 9 NO_ROOTS quadr_solver.h, 9 ONE_ROOT quadr_solver.h, 9 ONE_ROOT quadr_solver.h, 9 solve_quadr quadr_solver.cpp, 7 quadr_solver.cpp, 7 quadr_solver.h, 12 start_tests quadr_solver.h, 13 quadr_solver.cpp, 6 quadr_solver.h, 12 test_eq quadr_solver.h, 13 quadr_solver.h, 13 quadr_solver.h, 12 quadr_solver.h, 12 quadr_solver.h, 13 quadr_solver.h, 13 quadr_solver.h, 13 quadr_solver.h, 13 quadr_solver.h, 13		4
NO_ROOTS quadr_solver.h, 9 ONE_ROOT quadr_solver.h, 9 ONE_ROOT quadr_solver.h, 9 start_tests quadr_solver.h, 13 quadr_solver.cpp, 6 quadr_solver.h, 12 test_eq quadr_solver.h, 13 quadr_solver.h, 12 quadr_solver.h, 12 quadr_solver.h, 12 quadr_solver.h, 12 quadr_solver.h, 13 quadr_solver.h, 13 quadr_solver.h, 13 quadr_solver.h, 13	_	solve_lin
NO_ROOTS quadr_solver.h, 9 quadr_solver.h, 9 ONE_ROOT quadr_solver.h, 9 quadr_solver.h, 12 quadr_solver.h, 9 start_tests quadr_solver.h, 13 quadr_solver.cpp, 6 quadr_solver.h, 12 test_eq quadr_solver.h, 13 quadr_solver.h, 12 quadr_solver.h, 12 quadr_solver.h, 13 quadr_solver.h, 13 quadr_solver.h, 13 quadr_solver.h, 13	quadr_solver.n, 9	quadr solver.cpp, 7
quadr_solver.h, 9 quadr_solver.cpp, 7 quadr_solver.h, 12 quadr_solver.h, 9 start_tests quadr_solver.h, 13 quadr_solver.cpp, 6 quadr_solver.h, 12 quadr_solver.h, 12 quadr_solver.h, 12 quadr_solver.h, 13 quadr_solver.h, 13 quadr_solver.h, 13 quadr_solver.h, 13 quadr_solver.h, 13 quadr_solver.cpp, 3	NO POOTS	quadr solver.h, 12
quadr_solver.n, 9 Quadr_solver.cpp, 7 Quadr_solver.h, 12 quadr_solver.h, 9 start_tests quadr_solver.h, 13 quadr_solver.cpp, 6 quadr_solver.cpp, 6 quadr_solver.h, 12 quadr_solver.h, 12 quadr_solver.h, 13 quadr_solver.h, 13 quadr_solver.h, 13 quadr_solver.cpp, 3	_	• —
ONE_ROOT quadr_solver.h, 12 quadr_solver.h, 9 start_tests quadr_solver.h, 13 quadr_solver.cpp, 6 quadr_solver.h, 12 quadr_solver.h, 12 test_eq quadr_solver.h, 13 quadr_solver.h, 13 quadr_solver.h, 13 quadr_solver.cpp, 3	quadi_solver.n, 9	— ·
quadr_solver.h, 9 quadr_solver.h, 13 print_roots quadr_solver.cpp, 6 quadr_solver.h, 12 quadr_solver.h, 12 quadr_solver.h, 13 quadr_solver.h, 13 quadr_solver.cpp, 3	ONE BOOT	
quadr_solver.h, 13 print_roots quadr_tests.cpp, 14 quadr_solver.cpp, 6 quadr_solver.h, 12 test_eq quadr_solver.h, 13 quadr_solver.pp, 3 quadr_tests.cpp, 14	_	• —
print_roots quadr_tests.cpp, 14 quadr_solver.cpp, 6 quadr_solver.h, 12 test_eq quadr_solver.h, 13 quadr_solver.cpp, 3 quadr_tests.cpp, 14	quadi_borroim, b	-
quadr_solver.cpp, 6 quadr_solver.h, 12 test_eq quadr_solver.h, 13 quadr_solver.cpp, 3 quadr_tests.cpp, 14	print roots	• —
quadr_solver.h, 12 test_eq quadr_solver.h, 13 quadr_solver.cpp, 3 quadr_tests.cpp, 14	•	
quadr_solver.h, 13 quadr_solver.cpp, 3 quadr_tests.cpp, 14	. –	test_eq
quadr_solver.cpp, 3 quadr_tests.cpp, 14	1 , .=	_ ·
	quadr_solver.cpp, 3	
	are_doubles_equal, 4	TWO ROOTS
ask_for_coeff, 4 quadr_solver.h, 9		<u>–</u>
clear_buffer, 5		, , ,