

RPN

Generated by Doxygen 1.13.2



|  |           |
|--|-----------|
| <b>1 Namespace Index</b>                     | <b>1</b>  |
| 1.1 Namespace List                           | 1         |
| <b>2 Class Index</b>                         | <b>3</b>  |
| 2.1 Class List                               | 3         |
| <b>3 File Index</b>                          | <b>5</b>  |
| 3.1 File List                                | 5         |
| <b>4 Namespace Documentation</b>             | <b>7</b>  |
| 4.1 RPN Namespace Reference                  | 7         |
| 4.1.1 Function Documentation                 | 7         |
| 4.1.1.1 calculate() [1/2]                    | 7         |
| 4.1.1.2 calculate() [2/2]                    | 8         |
| 4.1.1.3 handleCbrt()                         | 8         |
| 4.1.1.4 handleDivision()                     | 8         |
| 4.1.1.5 handleSqrt()                         | 9         |
| 4.1.1.6 is1ArgOperator()                     | 9         |
| 4.1.1.7 is2ArgOperator()                     | 9         |
| 4.1.1.8 isOperator()                         | 9         |
| 4.1.1.9 sumLetters()                         | 10        |
| 4.1.2 Variable Documentation                 | 10        |
| 4.1.2.1 ADD_SUB_PREC                         | 10        |
| 4.1.2.2 EXP_PREC                             | 10        |
| 4.1.2.3 MULT_DIV_PREC                        | 10        |
| 4.1.2.4 one_arg_operators                    | 11        |
| 4.1.2.5 operatorPrecedence                   | 11        |
| 4.1.2.6 TRIG_FUN_PREC                        | 11        |
| 4.1.2.7 two_arg_operators                    | 11        |
| <b>5 Class Documentation</b>                 | <b>13</b> |
| 5.1 RPN::NotationConverter Struct Reference  | 13        |
| 5.1.1 Detailed Description                   | 13        |
| 5.1.2 Member Function Documentation          | 13        |
| 5.1.2.1 infixToRPN()                         | 13        |
| 5.1.2.2 RPNtoInfix()                         | 13        |
| 5.2 RPN::NotationDeterminer Struct Reference | 14        |
| 5.2.1 Member Function Documentation          | 14        |
| 5.2.1.1 isInfix()                            | 14        |
| 5.2.1.2 isRPN()                              | 14        |
| 5.3 RPN::RPNSolver Struct Reference          | 15        |
| 5.3.1 Detailed Description                   | 15        |
| 5.3.2 Member Function Documentation          | 15        |
| 5.3.2.1 getResult()                          | 15        |

|   |           |
|---|-----------|
| 5.4 RPN::TokenReader Struct Reference                                     | 15        |
| 5.4.1 Detailed Description  | 15        |
| 5.4.2 Constructor & Destructor Documentation                              | 15        |
| 5.4.2.1 TokenReader()   | 15        |
| 5.4.3 Member Function Documentation                                       | 16        |
| 5.4.3.1 finished()  | 16        |
| 5.4.3.2 getString()   | 16        |
| 5.4.3.3 next()  | 16        |
| 5.4.3.4 peek()  | 16        |
| <b>6 File Documentation</b>   | <b>17</b> |
| 6.1 build/CMakeFiles/3.30.5/CompilerIdC/CMakeCCompilerId.c File Reference | 17        |
| 6.1.1 Macro Definition Documentation                                      | 18        |
| 6.1.1.1 __has_include   | 18        |
| 6.1.1.2 ARCHITECTURE_ID   | 18        |
| 6.1.1.3 C_STD_11  | 18        |
| 6.1.1.4 C_STD_17  | 18        |
| 6.1.1.5 C_STD_23  | 18        |
| 6.1.1.6 C_STD_99  | 18        |
| 6.1.1.7 C_VERSION   | 18        |
| 6.1.1.8 COMPILER_ID   | 18        |
| 6.1.1.9 DEC   | 19        |
| 6.1.1.10 HEX  | 19        |
| 6.1.1.11 PLATFORM_ID  | 19        |
| 6.1.1.12 STRINGIFY  | 19        |
| 6.1.1.13 STRINGIFY_HELPER   | 19        |
| 6.1.2 Function Documentation  | 19        |
| 6.1.2.1 main()  | 19        |
| 6.1.3 Variable Documentation  | 20        |
| 6.1.3.1 info_arch   | 20        |
| 6.1.3.2 info_compiler   | 20        |
| 6.1.3.3 info_language_extensions_default                                  | 20        |
| 6.1.3.4 info_language_standard_default                                    | 20        |
| 6.1.3.5 info_platform   | 20        |
| 6.2 build/CMakeFiles/3.31.0/CompilerIdC/CMakeCCompilerId.c File Reference | 20        |
| 6.2.1 Macro Definition Documentation                                      | 21        |
| 6.2.1.1 __has_include   | 21        |
| 6.2.1.2 ARCHITECTURE_ID   | 21        |
| 6.2.1.3 C_STD_11  | 21        |
| 6.2.1.4 C_STD_17  | 21        |
| 6.2.1.5 C_STD_23  | 21        |
| 6.2.1.6 C_STD_99  | 21        |

|   |    |
|---|----|
| 6.2.1.7 C_VERSION . . . . .   | 21 |
| 6.2.1.8 COMPILER_ID . . . . .   | 22 |
| 6.2.1.9 DEC . . . . .   | 22 |
| 6.2.1.10 HEX . . . . .  | 22 |
| 6.2.1.11 PLATFORM_ID . . . . .  | 22 |
| 6.2.1.12 STRINGIFY . . . . .  | 22 |
| 6.2.1.13 STRINGIFY_HELPER . . . . .   | 22 |
| 6.2.2 Function Documentation . . . . .  | 23 |
| 6.2.2.1 main() . . . . .  | 23 |
| 6.2.3 Variable Documentation . . . . .  | 23 |
| 6.2.3.1 info_arch . . . . .   | 23 |
| 6.2.3.2 info_compiler . . . . .   | 23 |
| 6.2.3.3 info_language_extensions_default . . . . .  | 23 |
| 6.2.3.4 info_language_standard_default . . . . .  | 23 |
| 6.2.3.5 info_platform . . . . .   | 23 |
| 6.3 build/CMakeFiles/3.30.5/CompilerIdCXX/CMakeCXXCompilerId.cpp File Reference . . . . . | 24 |
| 6.3.1 Macro Definition Documentation . . . . .  | 24 |
| 6.3.1.1 __has_include . . . . .   | 24 |
| 6.3.1.2 ARCHITECTURE_ID . . . . .   | 24 |
| 6.3.1.3 COMPILER_ID . . . . .   | 25 |
| 6.3.1.4 CXX_STD . . . . .   | 25 |
| 6.3.1.5 CXX_STD_11 . . . . .  | 25 |
| 6.3.1.6 CXX_STD_14 . . . . .  | 25 |
| 6.3.1.7 CXX_STD_17 . . . . .  | 25 |
| 6.3.1.8 CXX_STD_20 . . . . .  | 25 |
| 6.3.1.9 CXX_STD_23 . . . . .  | 25 |
| 6.3.1.10 CXX_STD_98 . . . . .   | 25 |
| 6.3.1.11 DEC . . . . .  | 25 |
| 6.3.1.12 HEX . . . . .  | 26 |
| 6.3.1.13 PLATFORM_ID . . . . .  | 26 |
| 6.3.1.14 STRINGIFY . . . . .  | 26 |
| 6.3.1.15 STRINGIFY_HELPER . . . . .   | 26 |
| 6.3.2 Function Documentation . . . . .  | 26 |
| 6.3.2.1 main() . . . . .  | 26 |
| 6.3.3 Variable Documentation . . . . .  | 26 |
| 6.3.3.1 info_arch . . . . .   | 26 |
| 6.3.3.2 info_compiler . . . . .   | 27 |
| 6.3.3.3 info_language_extensions_default . . . . .  | 27 |
| 6.3.3.4 info_language_standard_default . . . . .  | 27 |
| 6.3.3.5 info_platform . . . . .   | 27 |
| 6.4 build/CMakeFiles/3.31.0/CompilerIdCXX/CMakeCXXCompilerId.cpp File Reference . . . . . | 27 |
| 6.4.1 Macro Definition Documentation . . . . .  | 28 |

|  |    |
|--|----|
| 6.4.1.1 <code>__has_include</code>   | 28 |
| 6.4.1.2 <code>ARCHITECTURE_ID</code>   | 28 |
| 6.4.1.3 <code>COMPILER_ID</code>   | 28 |
| 6.4.1.4 <code>CXX_STD</code>   | 28 |
| 6.4.1.5 <code>CXX_STD_11</code>  | 28 |
| 6.4.1.6 <code>CXX_STD_14</code>  | 29 |
| 6.4.1.7 <code>CXX_STD_17</code>  | 29 |
| 6.4.1.8 <code>CXX_STD_20</code>  | 29 |
| 6.4.1.9 <code>CXX_STD_23</code>  | 29 |
| 6.4.1.10 <code>CXX_STD_98</code>   | 29 |
| 6.4.1.11 <code>DEC</code>  | 29 |
| 6.4.1.12 <code>HEX</code>  | 29 |
| 6.4.1.13 <code>PLATFORM_ID</code>  | 30 |
| 6.4.1.14 <code>STRINGIFY</code>  | 30 |
| 6.4.1.15 <code>STRINGIFY_HELPER</code>   | 30 |
| 6.4.2 Function Documentation   | 30 |
| 6.4.2.1 <code>main()</code>  | 30 |
| 6.4.3 Variable Documentation   | 30 |
| 6.4.3.1 <code>info_arch</code>   | 30 |
| 6.4.3.2 <code>info_compiler</code>   | 30 |
| 6.4.3.3 <code>info_language_extensions_default</code>                          | 30 |
| 6.4.3.4 <code>info_language_standard_default</code>                            | 31 |
| 6.4.3.5 <code>info_platform</code>   | 31 |
| 6.5 <code>build/CMakeFiles/RPN.dir/main.cpp.obj.d</code> File Reference        | 31 |
| 6.6 <code>build/lib/CMakeFiles/RPN_LIB.dir/RPN.cpp.obj.d</code> File Reference | 31 |
| 6.7 <code>lib/RPN.cpp</code> File Reference                                    | 31 |
| 6.8 <code>lib/RPN.h</code> File Reference                                      | 32 |
| 6.9 <code>RPN.h</code>   | 33 |
| 6.10 <code>main.cpp</code> File Reference                                      | 33 |
| 6.10.1 Macro Definition Documentation  | 34 |
| 6.10.1.1 <code>DEBUG</code>  | 34 |
| 6.10.2 Function Documentation  | 34 |
| 6.10.2.1 <code>help()</code>   | 34 |
| 6.10.2.2 <code>main()</code>   | 34 |
| 6.10.2.3 <code>setFlags()</code>   | 34 |
| 6.10.2.4 <code>solveForOutput()</code>   | 34 |
| 6.10.3 Variable Documentation  | 34 |
| 6.10.3.1 <code>inputFilePos</code>   | 34 |
| 6.10.3.2 <code>isInteractive</code>  | 34 |
| 6.10.3.3 <code>isRPNOutput</code>  | 34 |
| 6.10.3.4 <code>outputFilePos</code>  | 34 |







# Chapter 1

## Namespace Index

### 1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

|                               |                   |
|-------------------------------|-------------------|
| <a href="#">RPN</a> . . . . . | <a href="#">7</a> |
|-------------------------------|-------------------|



## Chapter 2

# Class Index

### 2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

|   |    |
|---|----|
| <a href="#">RPN::NotationConverter</a>  | 13 |
| <a href="#">RPN::NotationDeterminer</a> | 14 |
| <a href="#">RPN::RPNSolver</a>          | 15 |
| <a href="#">RPN::TokenReader</a>        | 15 |



# Chapter 3

## File Index

### 3.1 File List

Here is a list of all files with brief descriptions:

|   |    |
|---|----|
| <a href="#">main.cpp</a>  | 33 |
| build/CMakeFiles/3.30.5/CompilerIdC/ <a href="#">CMakeCCompilerId.c</a>       | 17 |
| build/CMakeFiles/3.30.5/CompilerIdCXX/ <a href="#">CMakeCXXCompilerId.cpp</a> | 24 |
| build/CMakeFiles/3.31.0/CompilerIdC/ <a href="#">CMakeCCompilerId.c</a>       | 20 |
| build/CMakeFiles/3.31.0/CompilerIdCXX/ <a href="#">CMakeCXXCompilerId.cpp</a> | 27 |
| build/CMakeFiles/RPN.dir/ <a href="#">main.cpp.obj.d</a>                      | 31 |
| build/lib/CMakeFiles/RPN_LIB.dir/ <a href="#">RPN.cpp.obj.d</a>               | 31 |
| lib/ <a href="#">RPN.cpp</a>  | 31 |
| lib/ <a href="#">RPN.h</a>  | 32 |



## Chapter 4

# Namespace Documentation

### 4.1 RPN Namespace Reference

#### Classes

- struct [NotationConverter](#)
- struct [NotationDeterminer](#)
- struct [RPNSolver](#)
- struct [TokenReader](#)

#### Functions

- int [sumLetters](#) (const std::string &str)
- double [handleDivision](#) (const double &a, const double &b)
- double [handleSqrt](#) (const double &a)
- double [handleCbrt](#) (const double &a)
- double [calculate](#) (const double &a, const double &b, const std::string &op)
- double [calculate](#) (const double &a, const std::string &op)
- bool [isOperator](#) (const std::string &op)
- bool [is1ArgOperator](#) (const std::string &op)
- bool [is2ArgOperator](#) (const std::string &op)

#### Variables

- constexpr int [EXP\\_PREC](#) = 100
- constexpr int [TRIG\\_FUN\\_PREC](#) = [EXP\\_PREC](#)-1
- constexpr int [MULT\\_DIV\\_PREC](#) = [TRIG\\_FUN\\_PREC](#)-1
- constexpr int [ADD\\_SUB\\_PREC](#) = [MULT\\_DIV\\_PREC](#)-1
- const std::map< std::string, int > [operatorPrecedence](#)
- const std::unordered\_set< std::string > [one\\_arg\\_operators](#)
- const std::unordered\_set< std::string > [two\\_arg\\_operators](#)

#### 4.1.1 Function Documentation

##### 4.1.1.1 [calculate\(\)](#) [1/2]

```
double RPN::calculate (
    const double & a,
    const double & b,
    const std::string & op)
```

Given operator and operands calculates the result

**Parameters**

|           |               |
|-----------|---------------|
| <i>a</i>  | left operand  |
| <i>b</i>  | right operand |
| <i>op</i> | operator      |

**Returns**

result

**4.1.1.2 calculate() [2/2]**

```
double RPN::calculate (  
    const double & a,  
    const std::string & op)
```

Given operator and operand calculates the result

**Parameters**

|           |          |
|-----------|----------|
| <i>a</i>  | operand  |
| <i>op</i> | operator |

**Returns**

result

Integers found in cases of this switch come from the sum of ascii values of letters of the operators.

**4.1.1.3 handleCbrt()**

```
double RPN::handleCbrt (  
    const double & a)
```

Calculates cubic roots and errors on negative numbers.

**Parameters**

|          |  |
|----------|--|
| <i>a</i> |  |
|----------|--|

**Returns**

cbrt(a)

**4.1.1.4 handleDivision()**

```
double RPN::handleDivision (  
    const double & a,  
    const double & b)
```

Handler for division. Throws error on divisor = 0.



**Parameters**

|          |  |
|----------|--|
| <i>a</i> |  |
| <i>b</i> |  |

**Returns**

a/b

**4.1.1.5 handleSqrt()**

```
double RPN::handleSqrt (  
    const double & a)
```

Calculates square roots and errors on negative numbers.

**Parameters**

|          |  |
|----------|--|
| <i>a</i> |  |
|----------|--|

**Returns**

sqrt(a)

**4.1.1.6 is1ArgOperator()**

```
bool RPN::is1ArgOperator (  
    const std::string & op)
```

Checks if given token is an operator that takes only 1 argument, e.g. sqrt(x).

**Returns**

true if is 1 argument operator.

**4.1.1.7 is2ArgOperator()**

```
bool RPN::is2ArgOperator (  
    const std::string & op)
```

Checks if given token is an operator that takes 2 arguments, e.g. a + b.

**Returns**

true if is 2 argument operator.

**4.1.1.8 isOperator()**

```
bool RPN::isOperator (  
    const std::string & op)
```

Checks if given string is a valid operator

**Parameters**

|           |  |
|-----------|--|
| <i>op</i> |  |
|-----------|--|

**Returns**

true if string is an operator

**4.1.1.9 sumLetters()**

```
int RPN::sumLetters (
    const std::string & str)
```

Intermediate function used by calculate for 1 parameter operators. Sums ascii values of letters to determine which switch case use.

**Parameters**

|            |  |
|------------|--|
| <i>str</i> |  |
|------------|--|

**Returns**

Ascii sum of letters.

**4.1.2 Variable Documentation****4.1.2.1 ADD\_SUB\_PREC**

```
int RPN::ADD_SUB_PREC = MULT_DIV_PREC-1 [constexpr]
```

Addition/subtraction precedence score.

**4.1.2.2 EXP\_PREC**

```
int RPN::EXP_PREC = 100 [constexpr]
```

Exponential precedence score.

**4.1.2.3 MULT\_DIV\_PREC**

```
int RPN::MULT_DIV_PREC = TRIG_FUN_PREC-1 [constexpr]
```

Multiplication/division precedence score.

#### 4.1.2.4 one\_arg\_operators

```
const std::unordered_set<std::string> RPN::one_arg_operators
```

**Initial value:**

```
= {
    "sqrt",
    "cbrt",
    "sin",
    "cos",
    "tan",
}
```

Operators taking 1 parameter

#### 4.1.2.5 operatorPrecedence

```
const std::map<std::string, int> RPN::operatorPrecedence
```

**Initial value:**

```
= {
    {"^", EXP_PREC},
    {"sqrt", EXP_PREC},
    {"cbrt", EXP_PREC},
    {"sin", TRIG_FUN_PREC},
    {"cos", TRIG_FUN_PREC},
    {"tan", TRIG_FUN_PREC},
    {"*", MULT_DIV_PREC},
    {"/", MULT_DIV_PREC},
    {"\\", MULT_DIV_PREC},
    {"+", ADD_SUB_PREC},
    {"-", ADD_SUB_PREC},
}
```

Mapped precedences to operators.

#### 4.1.2.6 TRIG\_FUN\_PREC

```
int RPN::TRIG_FUN_PREC = EXP_PREC-1 [constexpr]
```

Trigonometric functions precedence score.

#### 4.1.2.7 two\_arg\_operators

```
const std::unordered_set<std::string> RPN::two_arg_operators
```

**Initial value:**

```
= {
    "^",
    "*",
    "/",
    "\\",
    "+",
    "-",
}
```

Operators taking 2 parameters



# Chapter 5

## Class Documentation

### 5.1 RPN::NotationConverter Struct Reference

```
#include <RPN.h>
```

#### Static Public Member Functions

- static std::string [infixToRPN](#) (const std::string &infix)
- static std::string [RPNtoInfix](#) (const std::string &RPN)

#### 5.1.1 Detailed Description

Struct able to convert Infix to [RPN](#) and vice versa.

#### 5.1.2 Member Function Documentation

##### 5.1.2.1 infixToRPN()

```
std::string RPN::NotationConverter::infixToRPN (  
    const std::string & infix) [static]
```

Given infix equation string, converts it into [RPN](#) equation.

#### Parameters

|              |                |
|--------------|----------------|
| <i>infix</i> | Infix equation |
|--------------|----------------|

#### Returns

[RPN](#) equation.

##### 5.1.2.2 RPNtoInfix()

```
std::string RPN::NotationConverter::RPNtoInfix (  
    const std::string & RPN) [static]
```

Given [RPN](#) equation string, converts it into infix equation.

## Parameters

|                     |                               |
|---------------------|-------------------------------|
| <a href="#">RPN</a> | <a href="#">RPN</a> equation. |
|---------------------|-------------------------------|

## Returns

Infix equation.

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

- lib/[RPN.h](#)
- lib/[RPN.cpp](#)

## 5.2 RPN::NotationDeterminer Struct Reference

```
#include <RPN.h>
```

### Static Public Member Functions

- static bool [isRPN](#) (const std::string &equation)
- static bool [isInfix](#) (const std::string &equation)

### 5.2.1 Member Function Documentation

#### 5.2.1.1 [isInfix\(\)](#)

```
bool RPN::NotationDeterminer::isInfix (  
    const std::string & equation) [static]
```

Determines if string is an Infix equation.

## Returns

true if equation is written in Infix.

#### 5.2.1.2 [isRPN\(\)](#)

```
bool RPN::NotationDeterminer::isRPN (  
    const std::string & equation) [static]
```

Determines if string is an [RPN](#) equation.

## Returns

true if equation is written in [RPN](#).

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

- lib/[RPN.h](#)
- lib/[RPN.cpp](#)

## 5.3 RPN::RPNSolver Struct Reference

```
#include <RPN.h>
```

### Static Public Member Functions

- static double [getResult](#) (const std::string &equation)

### 5.3.1 Detailed Description

[RPN](#) equation solver.

### 5.3.2 Member Function Documentation

#### 5.3.2.1 getResult()

```
double RPN::RPNSolver::getResult (
    const std::string & equation) [static]
```

Solves for the result of the [RPN](#) equation.

#### Returns

Result of the [RPN](#) equation.

Takes 2 tokens from the stack, removing the first and reassigning the second to the result of the operation.

After the entire algorithm is done the stack should contain only 1 token, which is equal to the result of the equation.

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

- lib/[RPN.h](#)
- lib/[RPN.cpp](#)

## 5.4 RPN::TokenReader Struct Reference

```
#include <RPN.h>
```

### Public Member Functions

- [TokenReader](#) (const std::string &string)
- std::string [getString](#) ()
- std::string [next](#) ()
- std::string [peek](#) ()
- bool [finished](#) () const

### 5.4.1 Detailed Description

Wrapper over std::stringstream for extracting tokens from the string.

### 5.4.2 Constructor & Destructor Documentation

#### 5.4.2.1 TokenReader()

```
RPN::TokenReader::TokenReader (
    const std::string & string) [explicit]
```

Token reader constructor

## Parameters

|               |   |
|---------------|---|
| <i>string</i> | Reference to the string from which tokens are read. |
|---------------|---|

## 5.4.3 Member Function Documentation

### 5.4.3.1 finished()

```
bool RPN::TokenReader::finished () const
```

Checks if stream came to an end.

## Returns

true if stream has finished.

### 5.4.3.2 getString()

```
std::string RPN::TokenReader::getString ()
```

Returns the entire string from which reader reads.

## Returns

Whole string

### 5.4.3.3 next()

```
std::string RPN::TokenReader::next ()
```

Next tokens in the stream.

## Returns

Next token.

### 5.4.3.4 peek()

```
std::string RPN::TokenReader::peek ()
```

Checks upcoming token in the stream, while keeping the current position in the stream.

## Returns

Upcoming token in the stream.

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

- [lib/RPN.h](#)
- [lib/RPN.cpp](#)



## Chapter 6

# File Documentation

### 6.1 build/CMakeFiles/3.30.5/CompilerIdC/CMakeCCompilerId.c File Reference

#### Macros

- `#define __has_include(x)`
- `#define COMPILER_ID ""`
- `#define STRINGIFY_HELPER(X)`
- `#define STRINGIFY(X)`
- `#define PLATFORM_ID`
- `#define ARCHITECTURE_ID`
- `#define DEC(n)`
- `#define HEX(n)`
- `#define C_STD_99 199901L`
- `#define C_STD_11 201112L`
- `#define C_STD_17 201710L`
- `#define C_STD_23 202311L`
- `#define C_VERSION`

#### Functions

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

#### Variables

- `char const * info_compiler = "INFO" ":" "compiler[" COMPILER_ID "]"`
- `char const * info_platform = "INFO" ":" "platform[" PLATFORM_ID "]"`
- `char const * info_arch = "INFO" ":" "arch[" ARCHITECTURE_ID "]"`
- `const char * info_language_standard_default`
- `const char * info_language_extensions_default`

## 6.1.1 Macro Definition Documentation

### 6.1.1.1 `__has_include`

```
#define __has_include(  
    x)
```

**Value:**

0

### 6.1.1.2 `ARCHITECTURE_ID`

```
#define ARCHITECTURE_ID
```

### 6.1.1.3 `C_STD_11`

```
#define C_STD_11 201112L
```

### 6.1.1.4 `C_STD_17`

```
#define C_STD_17 201710L
```

### 6.1.1.5 `C_STD_23`

```
#define C_STD_23 202311L
```

### 6.1.1.6 `C_STD_99`

```
#define C_STD_99 199901L
```

### 6.1.1.7 `C_VERSION`

```
#define C_VERSION
```

### 6.1.1.8 `COMPILER_ID`

```
#define COMPILER_ID ""
```

### 6.1.1.9 DEC

```
#define DEC(
    n)
```

**Value:**

```
('0' + ((n) / 10000000) % 10), \
('0' + ((n) / 1000000) % 10), \
('0' + ((n) / 100000) % 10), \
('0' + ((n) / 10000) % 10), \
('0' + ((n) / 1000) % 10), \
('0' + ((n) / 100) % 10), \
('0' + ((n) / 10) % 10), \
('0' + ((n) % 10))
```

### 6.1.1.10 HEX

```
#define HEX(
    n)
```

**Value:**

```
('0' + ((n) >> 28 & 0xF)), \
('0' + ((n) >> 24 & 0xF)), \
('0' + ((n) >> 20 & 0xF)), \
('0' + ((n) >> 16 & 0xF)), \
('0' + ((n) >> 12 & 0xF)), \
('0' + ((n) >> 8 & 0xF)), \
('0' + ((n) >> 4 & 0xF)), \
('0' + ((n) & 0xF))
```

### 6.1.1.11 PLATFORM\_ID

```
#define PLATFORM_ID
```

### 6.1.1.12 STRINGIFY

```
#define STRINGIFY(
    X)
```

**Value:**

```
STRINGIFY_HELPER(X)
```

### 6.1.1.13 STRINGIFY\_HELPER

```
#define STRINGIFY_HELPER(
    X)
```

**Value:**

```
#X
```

## 6.1.2 Function Documentation

### 6.1.2.1 main()

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

### 6.1.3 Variable Documentation

#### 6.1.3.1 info\_arch

```
char const* info_arch = "INFO" ":" "arch[" ARCHITECTURE_ID "]"
```

#### 6.1.3.2 info\_compiler

```
char const* info_compiler = "INFO" ":" "compiler[" COMPILER_ID "]"
```

#### 6.1.3.3 info\_language\_extensions\_default

```
const char* info_language_extensions_default
```

**Initial value:**

```
= "INFO" ":" "extensions_default["
```

```
    "OFF"
"]"
```

#### 6.1.3.4 info\_language\_standard\_default

```
const char* info_language_standard_default
```

**Initial value:**

```
=
  "INFO" ":" "standard_default[" C_VERSION "]"
```

#### 6.1.3.5 info\_platform

```
char const* info_platform = "INFO" ":" "platform[" PLATFORM_ID "]"
```

## 6.2 build/CMakeFiles/3.31.0/CompilerIdC/CMakeCCompilerId.c File Reference

### Macros

- `#define __has_include(x)`
- `#define COMPILER_ID ""`
- `#define STRINGIFY_HELPER(X)`
- `#define STRINGIFY(X)`
- `#define PLATFORM_ID`
- `#define ARCHITECTURE_ID`
- `#define DEC(n)`
- `#define HEX(n)`
- `#define C_STD_99 199901L`
- `#define C_STD_11 201112L`
- `#define C_STD_17 201710L`
- `#define C_STD_23 202311L`
- `#define C_VERSION`

## Functions

- int [main](#) (int argc, char \*argv[ ])

## Variables

- char const \* [info\\_compiler](#) = "INFO" ":" "compiler[" COMPILER\_ID "]"
- char const \* [info\\_platform](#) = "INFO" ":" "platform[" PLATFORM\_ID "]"
- char const \* [info\\_arch](#) = "INFO" ":" "arch[" ARCHITECTURE\_ID "]"
- const char \* [info\\_language\\_standard\\_default](#)
- const char \* [info\\_language\\_extensions\\_default](#)

## 6.2.1 Macro Definition Documentation

### 6.2.1.1 `__has_include`

```
#define __has_include(  
    x)
```

#### Value:

0

### 6.2.1.2 `ARCHITECTURE_ID`

```
#define ARCHITECTURE_ID
```

### 6.2.1.3 `C_STD_11`

```
#define C_STD_11 201112L
```

### 6.2.1.4 `C_STD_17`

```
#define C_STD_17 201710L
```

### 6.2.1.5 `C_STD_23`

```
#define C_STD_23 202311L
```

### 6.2.1.6 `C_STD_99`

```
#define C_STD_99 199901L
```

### 6.2.1.7 `C_VERSION`

```
#define C_VERSION
```

### 6.2.1.8 COMPILER\_ID

```
#define COMPILER_ID ""
```

### 6.2.1.9 DEC

```
#define DEC(  
    n)
```

#### Value:

```
('0' + ((n) / 10000000) % 10), \
('0' + ((n) / 1000000) % 10), \
('0' + ((n) / 100000) % 10), \
('0' + ((n) / 10000) % 10), \
('0' + ((n) / 1000) % 10), \
('0' + ((n) / 100) % 10), \
('0' + ((n) / 10) % 10), \
('0' + ((n) % 10))
```

### 6.2.1.10 HEX

```
#define HEX(  
    n)
```

#### Value:

```
('0' + ((n) >> 28 & 0xF)), \
('0' + ((n) >> 24 & 0xF)), \
('0' + ((n) >> 20 & 0xF)), \
('0' + ((n) >> 16 & 0xF)), \
('0' + ((n) >> 12 & 0xF)), \
('0' + ((n) >> 8 & 0xF)), \
('0' + ((n) >> 4 & 0xF)), \
('0' + ((n) & 0xF))
```

### 6.2.1.11 PLATFORM\_ID

```
#define PLATFORM_ID
```

### 6.2.1.12 STRINGIFY

```
#define STRINGIFY(  
    X)
```

#### Value:

```
STRINGIFY_HELPER(X)
```

### 6.2.1.13 STRINGIFY\_HELPER

```
#define STRINGIFY_HELPER(  
    X)
```

#### Value:

```
#X
```

## 6.2.2 Function Documentation

### 6.2.2.1 main()

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

## 6.2.3 Variable Documentation

### 6.2.3.1 info\_arch

```
char const* info_arch = "INFO" ":" "arch[" ARCHITECTURE_ID "]"
```

### 6.2.3.2 info\_compiler

```
char const* info_compiler = "INFO" ":" "compiler[" COMPILER_ID "]"
```

### 6.2.3.3 info\_language\_extensions\_default

```
const char* info_language_extensions_default
```

#### Initial value:

```
= "INFO" ":" "extensions_default["
```

```
    "OFF"  
"]"
```

### 6.2.3.4 info\_language\_standard\_default

```
const char* info_language_standard_default
```

#### Initial value:

```
=  
    "INFO" ":" "standard_default[" C_VERSION "]"
```

### 6.2.3.5 info\_platform

```
char const* info_platform = "INFO" ":" "platform[" PLATFORM_ID "]"
```

## 6.3 build/CMakeFiles/3.30.5/CompilerIdCXX/CMakeCXXCompilerId.cpp

### File Reference

#### Macros

- `#define __has_include(x)`
- `#define COMPILER_ID ""`
- `#define STRINGIFY_HELPER(X)`
- `#define STRINGIFY(X)`
- `#define PLATFORM_ID`
- `#define ARCHITECTURE_ID`
- `#define DEC(n)`
- `#define HEX(n)`
- `#define CXX_STD_98 199711L`
- `#define CXX_STD_11 201103L`
- `#define CXX_STD_14 201402L`
- `#define CXX_STD_17 201703L`
- `#define CXX_STD_20 202002L`
- `#define CXX_STD_23 202302L`
- `#define CXX_STD __cplusplus`

#### Functions

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

#### Variables

- `char const * info_compiler = "INFO" ":" "compiler[" COMPILER_ID "]"`
- `char const * info_platform = "INFO" ":" "platform[" PLATFORM_ID "]"`
- `char const * info_arch = "INFO" ":" "arch[" ARCHITECTURE_ID "]"`
- `const char * info_language_standard_default`
- `const char * info_language_extensions_default`

### 6.3.1 Macro Definition Documentation

#### 6.3.1.1 \_\_has\_include

```
#define __has_include(  
    x)
```

##### Value:

0

#### 6.3.1.2 ARCHITECTURE\_ID

```
#define ARCHITECTURE_ID
```



### 6.3.1.3 COMPILER\_ID

```
#define COMPILER_ID ""
```

### 6.3.1.4 CXX\_STD

```
#define CXX_STD __cplusplus
```

### 6.3.1.5 CXX\_STD\_11

```
#define CXX_STD_11 201103L
```

### 6.3.1.6 CXX\_STD\_14

```
#define CXX_STD_14 201402L
```

### 6.3.1.7 CXX\_STD\_17

```
#define CXX_STD_17 201703L
```

### 6.3.1.8 CXX\_STD\_20

```
#define CXX_STD_20 202002L
```

### 6.3.1.9 CXX\_STD\_23

```
#define CXX_STD_23 202302L
```

### 6.3.1.10 CXX\_STD\_98

```
#define CXX_STD_98 199711L
```

### 6.3.1.11 DEC

```
#define DEC(  
    n)
```

#### Value:

```
( '0' + ((n) / 10000000) % 10 ), \
( '0' + ((n) / 1000000) % 10 ), \
( '0' + ((n) / 100000) % 10 ), \
( '0' + ((n) / 10000) % 10 ), \
( '0' + ((n) / 1000) % 10 ), \
( '0' + ((n) / 100) % 10 ), \
( '0' + ((n) / 10) % 10 ), \
( '0' + ((n) % 10) )
```

### 6.3.1.12 HEX

```
#define HEX(  
    n)
```

#### Value:

```
( '0' + ((n)>>28 & 0xF) ), \
( '0' + ((n)>>24 & 0xF) ), \
( '0' + ((n)>>20 & 0xF) ), \
( '0' + ((n)>>16 & 0xF) ), \
( '0' + ((n)>>12 & 0xF) ), \
( '0' + ((n)>>8  & 0xF) ), \
( '0' + ((n)>>4  & 0xF) ), \
( '0' + ((n)      & 0xF) )
```

### 6.3.1.13 PLATFORM\_ID

```
#define PLATFORM_ID
```

### 6.3.1.14 STRINGIFY

```
#define STRINGIFY(  
    X)
```

#### Value:

```
STRINGIFY_HELPER(X)
```

### 6.3.1.15 STRINGIFY\_HELPER

```
#define STRINGIFY_HELPER(  
    X)
```

#### Value:

```
#X
```

## 6.3.2 Function Documentation

### 6.3.2.1 main()

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

## 6.3.3 Variable Documentation

### 6.3.3.1 info\_arch

```
char const* info_arch = "INFO" ":" "arch[" ARCHITECTURE_ID "]"
```

### 6.3.3.2 info\_compiler

```
char const* info_compiler = "INFO" ":" "compiler[" COMPILER_ID "]"
```

### 6.3.3.3 info\_language\_extensions\_default

```
const char* info_language_extensions_default
```

**Initial value:**

```
= "INFO" ":" "extensions_default["
```

```
    "OFF"  
"]"
```

### 6.3.3.4 info\_language\_standard\_default

```
const char* info_language_standard_default
```

**Initial value:**

```
= "INFO" ":" "standard_default["
```

```
    "98"  
"]"
```

### 6.3.3.5 info\_platform

```
char const* info_platform = "INFO" ":" "platform[" PLATFORM_ID "]"
```

## 6.4 build/CMakeFiles/3.31.0/CompilerIdCXX/CMakeCXXCompilerId.cpp File Reference

### Macros

- `#define __has_include(x)`
- `#define COMPILER_ID ""`
- `#define STRINGIFY_HELPER(X)`
- `#define STRINGIFY(X)`
- `#define PLATFORM_ID`
- `#define ARCHITECTURE_ID`

- `#define DEC(n)`
- `#define HEX(n)`
- `#define CXX_STD_98 199711L`
- `#define CXX_STD_11 201103L`
- `#define CXX_STD_14 201402L`
- `#define CXX_STD_17 201703L`
- `#define CXX_STD_20 202002L`
- `#define CXX_STD_23 202302L`
- `#define CXX_STD __cplusplus`

## Functions

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

## Variables

- `char const * info_compiler = "INFO" ":" "compiler[" COMPILER_ID "]"`
- `char const * info_platform = "INFO" ":" "platform[" PLATFORM_ID "]"`
- `char const * info_arch = "INFO" ":" "arch[" ARCHITECTURE_ID "]"`
- `const char * info_language_standard_default`
- `const char * info_language_extensions_default`

## 6.4.1 Macro Definition Documentation

### 6.4.1.1 `__has_include`

```
#define __has_include(
    x)
```

#### Value:

```
0
```

### 6.4.1.2 `ARCHITECTURE_ID`

```
#define ARCHITECTURE_ID
```

### 6.4.1.3 `COMPILER_ID`

```
#define COMPILER_ID ""
```

### 6.4.1.4 `CXX_STD`

```
#define CXX_STD __cplusplus
```

### 6.4.1.5 `CXX_STD_11`

```
#define CXX_STD_11 201103L
```

#### 6.4.1.6 CXX\_STD\_14

```
#define CXX_STD_14 201402L
```

#### 6.4.1.7 CXX\_STD\_17

```
#define CXX_STD_17 201703L
```

#### 6.4.1.8 CXX\_STD\_20

```
#define CXX_STD_20 202002L
```

#### 6.4.1.9 CXX\_STD\_23

```
#define CXX_STD_23 202302L
```

#### 6.4.1.10 CXX\_STD\_98

```
#define CXX_STD_98 199711L
```

#### 6.4.1.11 DEC

```
#define DEC(  
    n)
```

##### Value:

```
('0' + ((n) / 10000000) % 10), \
('0' + ((n) / 1000000) % 10), \
('0' + ((n) / 100000) % 10), \
('0' + ((n) / 10000) % 10), \
('0' + ((n) / 1000) % 10), \
('0' + ((n) / 100) % 10), \
('0' + ((n) / 10) % 10), \
('0' + ((n) % 10))
```

#### 6.4.1.12 HEX

```
#define HEX(  
    n)
```

##### Value:

```
('0' + ((n) >> 28 & 0xF)), \
('0' + ((n) >> 24 & 0xF)), \
('0' + ((n) >> 20 & 0xF)), \
('0' + ((n) >> 16 & 0xF)), \
('0' + ((n) >> 12 & 0xF)), \
('0' + ((n) >> 8 & 0xF)), \
('0' + ((n) >> 4 & 0xF)), \
('0' + ((n) & 0xF))
```

#### 6.4.1.13 PLATFORM\_ID

```
#define PLATFORM_ID
```

#### 6.4.1.14 STRINGIFY

```
#define STRINGIFY(  
    X)
```

**Value:**

```
STRINGIFY_HELPER(X)
```

#### 6.4.1.15 STRINGIFY\_HELPER

```
#define STRINGIFY_HELPER(  
    X)
```

**Value:**

```
#X
```

### 6.4.2 Function Documentation

#### 6.4.2.1 main()

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

### 6.4.3 Variable Documentation

#### 6.4.3.1 info\_arch

```
char const* info_arch = "INFO" ":" "arch[" ARCHITECTURE_ID "]"
```

#### 6.4.3.2 info\_compiler

```
char const* info_compiler = "INFO" ":" "compiler[" COMPILER_ID "]"
```

#### 6.4.3.3 info\_language\_extensions\_default

```
const char* info_language_extensions_default
```

**Initial value:**

```
= "INFO" ":" "extensions_default["
```

```
    "OFF"
```

```
    "]"
```

#### 6.4.3.4 info\_language\_standard\_default

```
const char* info_language_standard_default
```

##### Initial value:

```
= "INFO" ":" "standard_default["
```

```
"98"
```

```
"]"
```

#### 6.4.3.5 info\_platform

```
char const* info_platform = "INFO" ":" "platform[" PLATFORM_ID "]"
```

## 6.5 build/CMakeFiles/RPN.dir/main.cpp.obj.d File Reference

## 6.6 build/lib/CMakeFiles/RPN\_LIB.dir/RPN.cpp.obj.d File Reference

## 6.7 lib/RPN.cpp File Reference

```
#include <iostream>
#include <memory>
#include <sstream>
#include <string>
#include <stack>
#include <unordered_set>
#include <climits>
#include <cmath>
#include <map>
#include "RPN.h"
```

### Namespaces

- namespace [RPN](#)

## Functions

- int [RPN::sumLetters](#) (const std::string &str)
- double [RPN::handleDivision](#) (const double &a, const double &b)
- double [RPN::handleSqrt](#) (const double &a)
- double [RPN::handleCbrt](#) (const double &a)
- double [RPN::calculate](#) (const double &a, const double &b, const std::string &op)
- double [RPN::calculate](#) (const double &a, const std::string &op)
- bool [RPN::isOperator](#) (const std::string &op)
- bool [RPN::is1ArgOperator](#) (const std::string &op)
- bool [RPN::is2ArgOperator](#) (const std::string &op)

## Variables

- constexpr int [RPN::EXP\\_PREC](#) = 100
- constexpr int [RPN::TRIG\\_FUN\\_PREC](#) = [EXP\\_PREC](#)-1
- constexpr int [RPN::MULT\\_DIV\\_PREC](#) = [TRIG\\_FUN\\_PREC](#)-1
- constexpr int [RPN::ADD\\_SUB\\_PREC](#) = [MULT\\_DIV\\_PREC](#)-1
- const std::map< std::string, int > [RPN::operatorPrecedence](#)
- const std::unordered\_set< std::string > [RPN::one\\_arg\\_operators](#)
- const std::unordered\_set< std::string > [RPN::two\\_arg\\_operators](#)

## 6.8 lib/RPN.h File Reference

```
#include <string>
#include <unordered_set>
#include <sstream>
```

## Classes

- struct [RPN::TokenReader](#)
- struct [RPN::RPNSolver](#)
- struct [RPN::NotationConverter](#)
- struct [RPN::NotationDeterminer](#)

## Namespaces

- namespace [RPN](#)



## 6.9 RPN.h

[Go to the documentation of this file.](#)

```

00001 #pragma once
00002 #include <string>
00003 #include <unordered_set>
00004 #include <sstream>
00005
00006 namespace RPN {
00010     struct TokenReader {
00015         explicit TokenReader(const std::string& string);
00020         std::string getString();
00025         std::string next();
00031         std::string peek();
00036         bool finished() const;
00037     private:
00038         std::string string_;
00039         std::stringstream stream;
00040     };
00041
00045     struct RPNsolver {
00050         static double getResult(const std::string& equation);
00051     };
00052
00056     struct NotationConverter {
00062         static std::string infixToRPN(const std::string& infix);
00068         static std::string RPNtoInfix(const std::string& RPN);
00069     private:
00078         static std::string wrapInParentheses(const std::string& a, const std::string& b, const
std::string& op);
00087         static std::string aopb(const std::string &a, const std::string &b, const std::string &op);
00088         static std::string onlyParentheses(const std::string &a);
00089     };
00090
00091     struct NotationDeterminer {
00096         static bool isRPN(const std::string& equation);
00101         static bool isInfix(const std::string& equation);
00102     };
00103 }

```

## 6.10 main.cpp File Reference

```

#include <iostream>
#include <cstdio>
#include <fstream>
#include <string>
#include "RPN.h"

```

### Macros

- #define `DEBUG` 0

### Functions

- void `help` ()
- void `setFlags` (const std::string &flags)
- void `solveForOutput` (const std::string &sourceEquation, std::string &outputEquation, double &result)
- int `main` (const int argc, char \*argv[])

### Variables

- int `inputFilePos` = -1
- int `outputFilePos` = -1
- bool `isInteractive` = false
- bool `isRPNOutput` = false

## 6.10.1 Macro Definition Documentation

### 6.10.1.1 DEBUG

```
#define DEBUG 0
```

## 6.10.2 Function Documentation

### 6.10.2.1 help()

```
void help ()
```

Outputs help when executable has no parameters.

### 6.10.2.2 main()

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

### 6.10.2.3 setFlags()

```
void setFlags (
    const std::string & flags)
```

Reads flags and configures their values.

Parameters

|              |  |
|--------------|--|
| <i>flags</i> |  |
|--------------|--|

### 6.10.2.4 solveForOutput()

```
void solveForOutput (
    const std::string & sourceEquation,
    std::string & outputEquation,
    double & result)
```

## 6.10.3 Variable Documentation

### 6.10.3.1 inputFilePos

```
int inputFilePos = -1
```

### 6.10.3.2 isInteractive

```
bool isInteractive = false
```

### 6.10.3.3 isRPNOutput

```
bool isRPNOutput = false
```

### 6.10.3.4 outputFilePos

```
int outputFilePos = -1
```

# Index

- `__has_include`
    - `CMakeCCompilerId.c`, [18](#), [21](#)
    - `CMakeCXXCompilerId.cpp`, [24](#), [28](#)
- `ADD_SUB_PREC`
  - `RPN`, [10](#)
- `ARCHITECTURE_ID`
  - `CMakeCCompilerId.c`, [18](#), [21](#)
  - `CMakeCXXCompilerId.cpp`, [24](#), [28](#)
- `build/CMakeFiles/3.30.5/CompilerIdC/CMakeCCompilerId.c`,  
[17](#)
- `build/CMakeFiles/3.30.5/CompilerIdCXX/CMakeCXXCompilerId.cpp`,  
[24](#)
- `build/CMakeFiles/3.31.0/CompilerIdC/CMakeCCompilerId.c`,  
[20](#)
- `build/CMakeFiles/3.31.0/CompilerIdCXX/CMakeCXXCompilerId.cpp`,  
[27](#)
- `build/CMakeFiles/RPN.dir/main.cpp.obj.d`, [31](#)
- `build/lib/CMakeFiles/RPN_LIB.dir/RPN.cpp.obj.d`, [31](#)
- `C_STD_11`
  - `CMakeCCompilerId.c`, [18](#), [21](#)
- `C_STD_17`
  - `CMakeCCompilerId.c`, [18](#), [21](#)
- `C_STD_23`
  - `CMakeCCompilerId.c`, [18](#), [21](#)
- `C_STD_99`
  - `CMakeCCompilerId.c`, [18](#), [21](#)
- `C_VERSION`
  - `CMakeCCompilerId.c`, [18](#), [21](#)
- `calculate`
  - `RPN`, [7](#), [8](#)
- `CMakeCCompilerId.c`
  - `__has_include`, [18](#), [21](#)
  - `ARCHITECTURE_ID`, [18](#), [21](#)
  - `C_STD_11`, [18](#), [21](#)
  - `C_STD_17`, [18](#), [21](#)
  - `C_STD_23`, [18](#), [21](#)
  - `C_STD_99`, [18](#), [21](#)
  - `C_VERSION`, [18](#), [21](#)
  - `COMPILER_ID`, [18](#), [21](#)
  - `DEC`, [18](#), [22](#)
  - `HEX`, [19](#), [22](#)
  - `info_arch`, [20](#), [23](#)
  - `info_compiler`, [20](#), [23](#)
  - `info_language_extensions_default`, [20](#), [23](#)
  - `info_language_standard_default`, [20](#), [23](#)
  - `info_platform`, [20](#), [23](#)
  - `main`, [19](#), [23](#)
- `PLATFORM_ID`, [19](#), [22](#)
- `STRINGIFY`, [19](#), [22](#)
- `STRINGIFY_HELPER`, [19](#), [22](#)
- `CMakeCXXCompilerId.cpp`
  - `__has_include`, [24](#), [28](#)
  - `ARCHITECTURE_ID`, [24](#), [28](#)
  - `COMPILER_ID`, [24](#), [28](#)
  - `CXX_STD`, [25](#), [28](#)
  - `CXX_STD_11`, [25](#), [28](#)
  - `CXX_STD_14`, [25](#), [28](#)
  - `CXX_STD_17`, [25](#), [29](#)
  - `CXX_STD_20`, [25](#), [29](#)
  - `CXX_STD_23`, [25](#), [29](#)
  - `CXX_STD_98`, [25](#), [29](#)
  - `DEC`, [25](#), [29](#)
  - `HEX`, [25](#), [29](#)
  - `info_arch`, [26](#), [30](#)
  - `info_compiler`, [26](#), [30](#)
  - `info_language_extensions_default`, [27](#), [30](#)
  - `info_language_standard_default`, [27](#), [30](#)
  - `info_platform`, [27](#), [31](#)
  - `main`, [26](#), [30](#)
  - `PLATFORM_ID`, [26](#), [29](#)
  - `STRINGIFY`, [26](#), [30](#)
  - `STRINGIFY_HELPER`, [26](#), [30](#)
- `COMPILER_ID`
  - `CMakeCCompilerId.c`, [18](#), [21](#)
  - `CMakeCXXCompilerId.cpp`, [24](#), [28](#)
- `CXX_STD`
  - `CMakeCXXCompilerId.cpp`, [25](#), [28](#)
- `CXX_STD_11`
  - `CMakeCXXCompilerId.cpp`, [25](#), [28](#)
- `CXX_STD_14`
  - `CMakeCXXCompilerId.cpp`, [25](#), [28](#)
- `CXX_STD_17`
  - `CMakeCXXCompilerId.cpp`, [25](#), [29](#)
- `CXX_STD_20`
  - `CMakeCXXCompilerId.cpp`, [25](#), [29](#)
- `CXX_STD_23`
  - `CMakeCXXCompilerId.cpp`, [25](#), [29](#)
- `CXX_STD_98`
  - `CMakeCXXCompilerId.cpp`, [25](#), [29](#)
- `DEBUG`
  - `main.cpp`, [34](#)
- `DEC`
  - `CMakeCCompilerId.c`, [18](#), [22](#)
  - `CMakeCXXCompilerId.cpp`, [25](#), [29](#)
- `EXP_PREC`

- RPN, 10
- finished
  - RPN::TokenReader, 16
- getResult
  - RPN::RPNSolver, 15
- getString
  - RPN::TokenReader, 16
- handleCbrt
  - RPN, 8
- handleDivision
  - RPN, 8
- handleSqrt
  - RPN, 9
- help
  - main.cpp, 34
- HEX
  - CMakeCCompilerId.c, 19, 22
  - CMakeCXXCompilerId.cpp, 25, 29
- infixToRPN
  - RPN::NotationConverter, 13
- info\_arch
  - CMakeCCompilerId.c, 20, 23
  - CMakeCXXCompilerId.cpp, 26, 30
- info\_compiler
  - CMakeCCompilerId.c, 20, 23
  - CMakeCXXCompilerId.cpp, 26, 30
- info\_language\_extensions\_default
  - CMakeCCompilerId.c, 20, 23
  - CMakeCXXCompilerId.cpp, 27, 30
- info\_language\_standard\_default
  - CMakeCCompilerId.c, 20, 23
  - CMakeCXXCompilerId.cpp, 27, 30
- info\_platform
  - CMakeCCompilerId.c, 20, 23
  - CMakeCXXCompilerId.cpp, 27, 31
- inputFilePos
  - main.cpp, 34
- is1ArgOperator
  - RPN, 9
- is2ArgOperator
  - RPN, 9
- isInfix
  - RPN::NotationDeterminer, 14
- isInteractive
  - main.cpp, 34
- isOperator
  - RPN, 9
- isRPN
  - RPN::NotationDeterminer, 14
- isRPNOutput
  - main.cpp, 34
- lib/RPN.cpp, 31
- lib/RPN.h, 32, 33
- main
  - CMakeCCompilerId.c, 19, 23
  - CMakeCXXCompilerId.cpp, 26, 30
  - main.cpp, 34
  - DEBUG, 34
  - help, 34
  - inputFilePos, 34
  - isInteractive, 34
  - isRPNOutput, 34
  - main, 34
  - outputFilePos, 34
  - setFlags, 34
  - solveForOutput, 34
  - MULT\_DIV\_PREC
    - RPN, 10
  - next
    - RPN::TokenReader, 16
  - one\_arg\_operators
    - RPN, 10
  - operatorPrecedence
    - RPN, 11
  - outputFilePos
    - main.cpp, 34
  - peek
    - RPN::TokenReader, 16
  - PLATFORM\_ID
    - CMakeCCompilerId.c, 19, 22
    - CMakeCXXCompilerId.cpp, 26, 29
  - RPN, 7
    - ADD\_SUB\_PREC, 10
    - calculate, 7, 8
    - EXP\_PREC, 10
    - handleCbrt, 8
    - handleDivision, 8
    - handleSqrt, 9
    - is1ArgOperator, 9
    - is2ArgOperator, 9
    - isOperator, 9
    - MULT\_DIV\_PREC, 10
    - one\_arg\_operators, 10
    - operatorPrecedence, 11
    - sumLetters, 10
    - TRIG\_FUN\_PREC, 11
    - two\_arg\_operators, 11
    - RPN::NotationConverter, 13
      - infixToRPN, 13
      - RPNtoInfix, 13
    - RPN::NotationDeterminer, 14
      - isInfix, 14
      - isRPN, 14
    - RPN::RPNSolver, 15
      - getResult, 15
    - RPN::TokenReader, 15
      - finished, 16
      - getString, 16

- next, [16](#)
  - peek, [16](#)
  - TokenReader, [15](#)
- RPNtoInfix
  - RPN::NotationConverter, [13](#)
- setFlags
  - main.cpp, [34](#)
- solveForOutput
  - main.cpp, [34](#)
- STRINGIFY
  - CMakeCCompilerId.c, [19](#), [22](#)
  - CMakeCXXCompilerId.cpp, [26](#), [30](#)
- STRINGIFY\_HELPER
  - CMakeCCompilerId.c, [19](#), [22](#)
  - CMakeCXXCompilerId.cpp, [26](#), [30](#)
- sumLetters
  - RPN, [10](#)
- TokenReader
  - RPN::TokenReader, [15](#)
- TRIG\_FUN\_PREC
  - RPN, [11](#)
- two\_arg\_operators
  - RPN, [11](#)