My_CPP_Project

1.0

Generated by Doxygen 1.15.0

Chapter 1

Directory Hierarchy

1.1 Directories

Chapter 2

File Index

2.1 File List

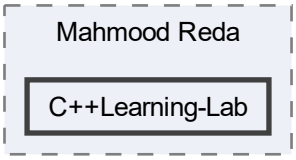
Here is a list of all files with brief descriptions:

Chapter 3

Directory Documentation

3.1 C++Learning-Lab Directory Reference

Directory dependency graph for C++Learning-Lab:



Files

- file main.cpp
- file main.h

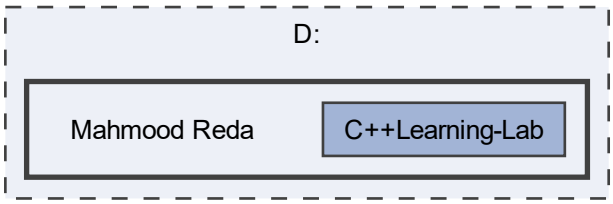
3.2 D: Directory Reference

Directories

- directory Mahmood Reda

3.3 Mahmood Reda Directory Reference

Directory dependency graph for Mahmood Reda:



Directories

- directory C++Learning-Lab

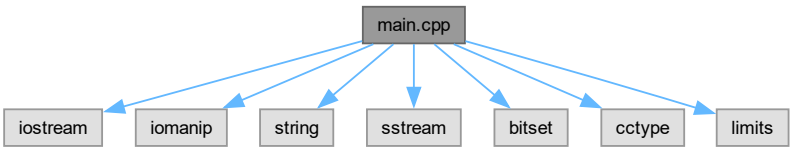
Chapter 4

File Documentation

4.1 main.cpp File Reference

```
#include <iostream>
#include <iomanip>
#include <string>
#include <sstream>
#include <bitset>
#include <cctype>
#include <limits>
```

Include dependency graph for main.cpp:



Functions

- void task_ascii_table ()

  Print a simple ASCII table for a user-specified range.
- void task_max_of_three ()

  Read three integers from the user and print the maximum.
- void task_right_triangle ()

  Print a right-angled triangle of '∗' characters.
- void task_check_vowel ()

  Check whether a single-character input is a vowel.
- void task_multiplication_table ()

  Print the multiplication table from 1 to 10.
- void task_sum_digits ()

  Calculate and print the sum of digits of a user-entered integer.
- void task_decimal_binary ()

  Convert decimal to binary and binary to decimal interactively.
- int main ()

  Display menu and run selected task until user chooses to exit.

4.1.1 Function Documentation

4.1.1.1 main()

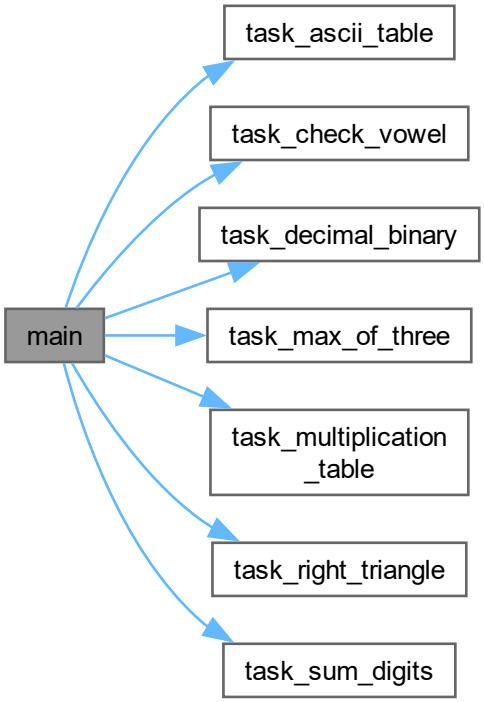
```
int main ()
```

Display menu and run selected task until user chooses to exit.

Definition at line 146 of file main.cpp.

```
00146           {
00147     while (true) {
00148         std::cout « "\n=== Tasks Menu ===\n"
00149                   « "1. ASCII table\n"
00150                   « "2. Maximum of three values\n"
00151                   « "3. RIGHT angle triangle\n"
00152                   « "4. Check if letter is vowel\n"
00153                   « "5. Multiplication table\n"
00154                   « "6. Sum digits of integer\n"
00155                   « "7. Decimal <-> Binary conversion\n"
00156                   « "0. Exit\n"
00157                   « "Select a task (0-7): ";
00158
00159         int choice;
00160         if (!(std::cin » choice)) {
00161             std::cin.clear();
00162             std::cin.ignore(std::numeric_limits<std::streamsize>::max(), '\n');
00163             std::cout « "Invalid input. Please enter a number between 0 and 7.\n";
00164             continue;
00165         }
00166
00167         switch (choice) {
00168             case 0: std::cout « "Exiting.\n"; return 0;
00169             case 1: task_ascii_table(); break;
00170             case 2: task_max_of_three(); break;
00171             case 3: task_right_triangle(); break;
00172             case 4: task_check_vowel(); break;
00173             case 5: task_multiplication_table(); break;
00174             case 6: task_sum_digits(); break;
00175             case 7: task_decimal_binary(); break;
00176             default: std::cout « "Please choose a valid option (0-7).\n"; break;
00177         }
00178     }
00179     return 0;
00180 }
```

Here is the call graph for this function:



4.1.1.2 task_ascii_table()

```
void task_ascii_table ()
```

Print a simple ASCII table for a user-specified range.

Prompts user for start and end codes (0-255) and prints Dec, Hex, Oct and Char (or label).

Definition at line 21 of file main.cpp.

```
00021                           {
00022     // Print table header
00023     std::cout « "ASCII Table\n";
00024     std::cout « "----------------------------\n";
00025     std::cout « "Char\t|\tASCII\n";
00026     std::cout « "----------------------------\n";
00027
00028     // Loop through ASCII codes from 0 to 240
00029     for (int i = 0; i <= 240; ++i) {
00030         std::cout « " Char " « char(i) « "\t|\t ASCII  " « i « std::endl;
00031     }
00032 }
```

Here is the caller graph for this function:

4.1.1.3 task_check_vowel()

```
void task_check_vowel ()
```

Check whether a single-character input is a vowel.

Validates input is an alphabetic character; compares against a,e,i,o,u (case-insensitive).

Definition at line 68 of file main.cpp.

```
00068                           {
00069     char letter;
00070     std::cout « "Enter a letter: ";
00071     if (!(std::cin » letter)) { std::cin.clear();
      std::cin.ignore(std::numeric_limits<std::streamsize>::max(), '\n'); return; }
00072
00073     if (!std::isalpha(static_cast<unsigned char>(letter))) {
00074         std::cout « "Invalid input. Please enter an alphabetic character.\n";
00075         return;
00076     }
00077     letter = static_cast<char>(std::tolower(static_cast<unsigned char>(letter)));
00078     if (letter == 'a' || letter == 'e' || letter == 'i' || letter == 'o' || letter == 'u')
00079         std::cout « "The letter '" « letter « "' is a vowel.\n";
00080     else
00081         std::cout « "The letter '" « letter « "' is not a vowel.\n";
00082 }
```

Here is the caller graph for this function:

4.1.1.4 task_decimal_binary()

```
void task_decimal_binary ()
```

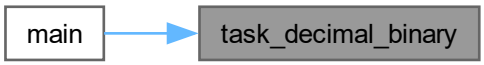
Convert decimal to binary and binary to decimal interactively.

Uses 8-bit bitset for decimal->binary and std::stoi with base 2 for binary->decimal.

Definition at line 117 of file main.cpp.

```
00117                           {
00118     int decimal;
00119     std::string binary;
00120     std::cout « "Enter a decimal number: ";
00121     if (!(std::cin » decimal)) { std::cin.clear();
      std::cin.ignore(std::numeric_limits<std::streamsize>::max(), '\n'); return; }
00122     std::bitset<16> bset(decimal); // choose 16 bits to be a bit more flexible
00123     std::cout « "Decimal number: " « decimal « '\n';
00124     std::cout « "Binary representation: " « bset.to_string() « '\n';
00125
00126     std::cout « "Enter a binary number: ";
00127     if (!(std::cin » binary)) { std::cin.clear();
      std::cin.ignore(std::numeric_limits<std::streamsize>::max(), '\n'); return; }
00128     // validate binary string
00129     for (char ch : binary) {
00130         if (ch != '0' && ch != '1') {
00131             std::cout « "Invalid binary input.\n";
00132             return;
00133         }
00134     }
00135     try {
00136         int dec = static_cast<int>(std::stol(binary, nullptr, 2));
00137         std::cout « "Decimal representation: " « dec « '\n';
00138     } catch (...) {
00139         std::cout « "Conversion error (number too large?)\n";
00140     }
00141 }
```

Here is the caller graph for this function:



4.1.1.5 task_max_of_three()

```
void task_max_of_three ()
```

Read three integers from the user and print the maximum.

Handles equal values by reporting that there is no single largest number.

Definition at line 38 of file main.cpp.

```
00038                           {
00039     int a=0, b=0, c=0;
00040     std::cout « "Enter first number: ";  if (!(std::cin » a)) { std::cin.clear();
      std::cin.ignore(std::numeric_limits<std::streamsize>::max(), '\n'); return; }
00041     std::cout « "Enter second number: "; if (!(std::cin » b)) { std::cin.clear();
      std::cin.ignore(std::numeric_limits<std::streamsize>::max(), '\n'); return; }
00042     std::cout « "Enter third number: ";  if (!(std::cin » c)) { std::cin.clear();
      std::cin.ignore(std::numeric_limits<std::streamsize>::max(), '\n'); return; }
00043
00044     if (a > b && a > c) std::cout « "The largest number is: " « a « '\n';
00045     else if (b > a && b > c) std::cout « "The largest number is: " « b « '\n';
00046     else if (c > a && c > b) std::cout « "The largest number is: " « c « '\n';
00047     else std::cout « "There is no single largest number (some numbers may be equal)." « '\n';
00048 }
```

Here is the caller graph for this function:

4.1.1.6 task_multiplication_table()

```
void task_multiplication_table ()
```

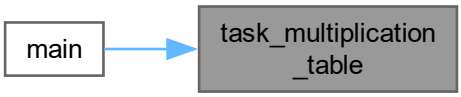
Print the multiplication table from 1 to 10.

Each product is printed in a grid-like row for readability.

Definition at line 88 of file main.cpp.

```
00088                                   {
00089     for (int i = 1; i <= 10; ++i) {
00090         for (int j = 1; j <= 10; ++j) {
00091             std::cout « std::setw(2) « i « "x" « std::setw(2) « j « "=" « std::setw(3) « (i*j) « "  ";
00092         }
00093         std::cout « '\n';
00094     }
00095 }
```

Here is the caller graph for this function:



4.1.1.7 task_right_triangle()

```
void task_right_triangle ()
```

Print a right-angled triangle of '*' characters.

Prompts the user for the number of rows; prints rows from 1..n stars.

Definition at line 54 of file main.cpp.

```
00054                                   {
00055     int rows = 0;
00056     std::cout « "Enter the number of rows: ";
```

```
00057     if (!(std::cin » rows)) { std::cin.clear();
      std::cin.ignore(std::numeric_limits<std::streamsize>::max(), '\n'); return; }
00058     for (int i = 1; i <= rows; ++i) {
00059         for (int j = 0; j < i; ++j) std::cout « '*';
00060         std::cout « '\n';
00061     }
00062 }
```

Here is the caller graph for this function:

4.1.1.8 task_sum_digits()

```
void task_sum_digits ()
```

Calculate and print the sum of digits of a user-entered integer.

Works with negative numbers by ignoring the sign.

Definition at line 101 of file main.cpp.

```
00101                       {
00102     int number;
00103     std::cout « "Enter a number: ";
00104     if (!(std::cin » number)) { std::cin.clear();
      std::cin.ignore(std::numeric_limits<std::streamsize>::max(), '\n'); return; }
00105     std::string s = std::to_string(number);
00106     int sum = 0;
00107     for (char ch : s) {
00108         if (std::isdigit(static_cast<unsigned char>(ch))) sum += (ch - '0');
00109     }
00110     std::cout « "Sum of digits = " « sum « '\n';
00111 }
```

Here is the caller graph for this function:

4.2 main.cpp

Go to the documentation of this file.

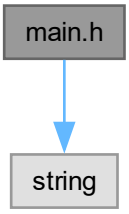
```
00001
00008
00009 #include <iostream>
00010 #include <iomanip>
00011 #include <string>
00012 #include <sstream>
00013 #include <bitset>
00014 #include <cctype>
00015 #include <limits>
00016
00021 void task_ascii_table() {
00022     // Print table header
00023     std::cout « "ASCII Table\n";
00024     std::cout « "-----------------------------\n";
00025     std::cout « "Char\t|\tASCII\n";
00026     std::cout « "-----------------------------\n";
00027
00028     // Loop through ASCII codes from 0 to 240
00029     for (int i = 0; i <= 240; ++i) {
00030         std::cout « " Char " « char(i) « "\t|\t ASCII  " « i « std::endl;
00031     }
00032 }
00033
00038 void task_max_of_three() {
00039     int a=0, b=0, c=0;
00040     std::cout « "Enter first number: ";  if (!(std::cin » a)) { std::cin.clear();
    std::cin.ignore(std::numeric_limits<std::streamsize>::max(), '\n'); return; }
00041     std::cout « "Enter second number: "; if (!(std::cin » b)) { std::cin.clear();
    std::cin.ignore(std::numeric_limits<std::streamsize>::max(), '\n'); return; }
00042     std::cout « "Enter third number: ";  if (!(std::cin » c)) { std::cin.clear();
    std::cin.ignore(std::numeric_limits<std::streamsize>::max(), '\n'); return; }
00043
00044     if (a > b && a > c) std::cout « "The largest number is: " « a « '\n';
00045     else if (b > a && b > c) std::cout « "The largest number is: " « b « '\n';
00046     else if (c > a && c > b) std::cout « "The largest number is: " « c « '\n';
00047     else std::cout « "There is no single largest number (some numbers may be equal)." « '\n';
00048 }
00049
00054 void task_right_triangle() {
00055     int rows = 0;
00056     std::cout « "Enter the number of rows: ";
00057     if (!(std::cin » rows)) { std::cin.clear();
    std::cin.ignore(std::numeric_limits<std::streamsize>::max(), '\n'); return; }
00058     for (int i = 1; i <= rows; ++i) {
00059         for (int j = 0; j < i; ++j) std::cout « '*';
00060         std::cout « '\n';
00061     }
00062 }
00063
00068 void task_check_vowel() {
00069     char letter;
00070     std::cout « "Enter a letter: ";
00071     if (!(std::cin » letter)) { std::cin.clear();
    std::cin.ignore(std::numeric_limits<std::streamsize>::max(), '\n'); return; }
00072
00073     if (!std::isalpha(static_cast<unsigned char>(letter))) {
00074         std::cout « "Invalid input. Please enter an alphabetic character.\n";
00075         return;
00076     }
00077     letter = static_cast<char>(std::tolower(static_cast<unsigned char>(letter)));
00078     if (letter == 'a' || letter == 'e' || letter == 'i' || letter == 'o' || letter == 'u')
00079         std::cout « "The letter '" « letter « "' is a vowel.\n";
00080     else
00081         std::cout « "The letter '" « letter « "' is not a vowel.\n";
00082 }
00083
00088 void task_multiplication_table() {
00089     for (int i = 1; i <= 10; ++i) {
00090         for (int j = 1; j <= 10; ++j) {
00091             std::cout « std::setw(2) « i « "x" « std::setw(2) « j « "=" « std::setw(3) « (i*j) « "  ";
00092         }
00093         std::cout « '\n';
00094     }
00095 }
00096
00101 void task_sum_digits() {
00102     int number;
00103     std::cout « "Enter a number: ";
00104     if (!(std::cin » number)) { std::cin.clear();
    std::cin.ignore(std::numeric_limits<std::streamsize>::max(), '\n'); return; }
00105     std::string s = std::to_string(number);
00106     int sum = 0;
```

```
00107     for (char ch : s) {
00108         if (std::isdigit(static_cast<unsigned char>(ch))) sum += (ch - '0');
00109     }
00110     std::cout « "Sum of digits = " « sum « '\n';
00111 }
00112
00117 void task_decimal_binary() {
00118     int decimal;
00119     std::string binary;
00120     std::cout « "Enter a decimal number: ";
00121     if (!(std::cin » decimal)) { std::cin.clear();
      std::cin.ignore(std::numeric_limits<std::streamsize>::max(), '\n'); return; }
00122     std::bitset<16> bset(decimal); // choose 16 bits to be a bit more flexible
00123     std::cout « "Decimal number: " « decimal « '\n';
00124     std::cout « "Binary representation: " « bset.to_string() « '\n';
00125
00126     std::cout « "Enter a binary number: ";
00127     if (!(std::cin » binary)) { std::cin.clear();
      std::cin.ignore(std::numeric_limits<std::streamsize>::max(), '\n'); return; }
00128     // validate binary string
00129     for (char ch : binary) {
00130         if (ch != '0' && ch != '1') {
00131             std::cout « "Invalid binary input.\n";
00132             return;
00133         }
00134     }
00135     try {
00136         int dec = static_cast<int>(std::stol(binary, nullptr, 2));
00137         std::cout « "Decimal representation: " « dec « '\n';
00138     } catch (...) {
00139         std::cout « "Conversion error (number too large?)\n";
00140     }
00141 }
00142
00146 int main() {
00147     while (true) {
00148         std::cout « "\n=== Tasks Menu ===\n"
00149                   « "1. ASCII table\n"
00150                   « "2. Maximum of three values\n"
00151                   « "3. RIGHT angle triangle\n"
00152                   « "4. Check if letter is vowel\n"
00153                   « "5. Multiplication table\n"
00154                   « "6. Sum digits of integer\n"
00155                   « "7. Decimal <-> Binary conversion\n"
00156                   « "0. Exit\n"
00157                   « "Select a task (0-7): ";
00158
00159         int choice;
00160         if (!(std::cin » choice)) {
00161             std::cin.clear();
00162             std::cin.ignore(std::numeric_limits<std::streamsize>::max(), '\n');
00163             std::cout « "Invalid input. Please enter a number between 0 and 7.\n";
00164             continue;
00165         }
00166
00167         switch (choice) {
00168             case 0: std::cout « "Exiting.\n"; return 0;
00169             case 1: task_ascii_table(); break;
00170             case 2: task_max_of_three(); break;
00171             case 3: task_right_triangle(); break;
00172             case 4: task_check_vowel(); break;
00173             case 5: task_multiplication_table(); break;
00174             case 6: task_sum_digits(); break;
00175             case 7: task_decimal_binary(); break;
00176             default: std::cout « "Please choose a valid option (0-7).\n"; break;
00177         }
00178     }
00179     return 0;
00180 }
```

4.3 main.h File Reference

```
#include <string>
```

Include dependency graph for main.h:



Functions

- void task_ascii_table ()

  Print a simple ASCII table for a user-specified range.
- void task_max_of_three ()

  Read three integers from the user and print the maximum.
- void task_right_triangle ()

  Print a right-angled triangle of '∗' characters.
- void task_check_vowel ()

  Check whether a single-character input is a vowel.
- void task_multiplication_table ()

  Print the multiplication table from 1 to 10.
- void task_sum_digits ()

  Calculate and print the sum of digits of a user-entered integer.
- void task_decimal_binary ()

  Convert decimal to binary and binary to decimal interactively.

4.3.1 Function Documentation

4.3.1.1 task_ascii_table()

```
void task_ascii_table ()
```

Print a simple ASCII table for a user-specified range.

Prompts user for start and end codes (0-255) and prints Dec, Hex, Oct and Char (or a label for non-printable codes).

Prompts user for start and end codes (0-255) and prints Dec, Hex, Oct and Char (or label).

Definition at line 21 of file main.cpp.

```
00021                           {
```

```
00022     // Print table header
00023     std::cout « "ASCII Table\n";
00024     std::cout « "----------------------------\n";
00025     std::cout « "Char\t|\tASCII\n";
00026     std::cout « "----------------------------\n";
00027
00028     // Loop through ASCII codes from 0 to 240
00029     for (int i = 0; i <= 240; ++i) {
00030         std::cout « " Char " « char(i) « "\t|\t ASCII  " « i « std::endl;
00031     }
00032 }
```

Here is the caller graph for this function:

4.3.1.2 task_check_vowel()

```
void task_check_vowel ()
```

Check whether a single-character input is a vowel.

Validates input is an alphabetic character; compares against a, e, i, o, u in a case-insensitive manner.

Validates input is an alphabetic character; compares against a,e,i,o,u (case-insensitive).

Definition at line 68 of file main.cpp.

```
00068                           {
00069     char letter;
00070     std::cout « "Enter a letter: ";
00071     if (!(std::cin » letter)) { std::cin.clear();
      std::cin.ignore(std::numeric_limits<std::streamsize>::max(), '\n'); return; }
00072
00073     if (!std::isalpha(static_cast<unsigned char>(letter))) {
00074         std::cout « "Invalid input. Please enter an alphabetic character.\n";
00075         return;
00076     }
00077     letter = static_cast<char>(std::tolower(static_cast<unsigned char>(letter)));
00078     if (letter == 'a' || letter == 'e' || letter == 'i' || letter == 'o' || letter == 'u')
00079         std::cout « "The letter '" « letter « "' is a vowel.\n";
00080     else
00081         std::cout « "The letter '" « letter « "' is not a vowel.\n";
00082 }
```

Here is the caller graph for this function:

4.3.1.3 task_decimal_binary()

```
void task_decimal_binary ()
```

Convert decimal to binary and binary to decimal interactively.

Uses a bitset for decimal->binary representation and std::stol with base 2 for binary->decimal conversion. Validates binary input.

Uses 8-bit bitset for decimal->binary and std::stoi with base 2 for binary->decimal.

Definition at line 117 of file main.cpp.

```
00117                           {
00118     int decimal;
00119     std::string binary;
00120     std::cout « "Enter a decimal number: ";
00121     if (!(std::cin » decimal)) { std::cin.clear();
      std::cin.ignore(std::numeric_limits<std::streamsize>::max(), '\n'); return; }
00122     std::bitset<16> bset(decimal); // choose 16 bits to be a bit more flexible
00123     std::cout « "Decimal number: " « decimal « '\n';
00124     std::cout « "Binary representation: " « bset.to_string() « '\n';
00125
00126     std::cout « "Enter a binary number: ";
00127     if (!(std::cin » binary)) { std::cin.clear();
      std::cin.ignore(std::numeric_limits<std::streamsize>::max(), '\n'); return; }
00128     // validate binary string
00129     for (char ch : binary) {
00130         if (ch != '0' && ch != '1') {
00131             std::cout « "Invalid binary input.\n";
00132             return;
00133         }
00134     }
00135     try {
00136         int dec = static_cast<int>(std::stol(binary, nullptr, 2));
00137         std::cout « "Decimal representation: " « dec « '\n';
00138     } catch (...) {
00139         std::cout « "Conversion error (number too large?)\n";
00140     }
00141 }
```

Here is the caller graph for this function:

4.3.1.4 task_max_of_three()

```
void task_max_of_three ()
```

Read three integers from the user and print the maximum.

Handles equal values by reporting that there is no single largest number.

Definition at line 38 of file main.cpp.

```
00038                           {
00039     int a=0, b=0, c=0;
00040     std::cout « "Enter first number: ";  if (!(std::cin » a)) { std::cin.clear();
      std::cin.ignore(std::numeric_limits<std::streamsize>::max(), '\n'); return; }
```

```
00041     std::cout « "Enter second number: "; if (!(std::cin » b)) { std::cin.clear();
      std::cin.ignore(std::numeric_limits<std::streamsize>::max(), '\n'); return; }
00042     std::cout « "Enter third number: ";  if (!(std::cin » c)) { std::cin.clear();
      std::cin.ignore(std::numeric_limits<std::streamsize>::max(), '\n'); return; }
00043
00044     if (a > b && a > c) std::cout « "The largest number is: " « a « '\n';
00045     else if (b > a && b > c) std::cout « "The largest number is: " « b « '\n';
00046     else if (c > a && c > b) std::cout « "The largest number is: " « c « '\n';
00047     else std::cout « "There is no single largest number (some numbers may be equal)." « '\n';
00048 }
```

Here is the caller graph for this function:

4.3.1.5 task_multiplication_table()

```
void task_multiplication_table ()
```

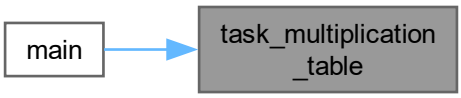
Print the multiplication table from 1 to 10.

Each product is printed in a grid-like row for readability.

Definition at line 88 of file main.cpp.

```
00088                                {
00089     for (int i = 1; i <= 10; ++i) {
00090         for (int j = 1; j <= 10; ++j) {
00091             std::cout « std::setw(2) « i « "x" « std::setw(2) « j « "=" « std::setw(3) « (i*j) « "  ";
00092         }
00093         std::cout « '\n';
00094     }
00095 }
```

Here is the caller graph for this function:



4.3.1.6 task_right_triangle()

```
void task_right_triangle ()
```

Print a right-angled triangle of '*' characters.

Prompts the user for the number of rows; prints rows from 1..n stars.

Definition at line 54 of file main.cpp.

```
00054                           {
00055     int rows = 0;
00056     std::cout « "Enter the number of rows: ";
00057     if (!(std::cin » rows)) { std::cin.clear();
      std::cin.ignore(std::numeric_limits<std::streamsize>::max(), '\n'); return; }
00058     for (int i = 1; i <= rows; ++i) {
00059         for (int j = 0; j < i; ++j) std::cout « '*';
00060         std::cout « '\n';
00061     }
00062 }
```

Here is the caller graph for this function:

4.3.1.7 task_sum_digits()

```
void task_sum_digits ()
```

Calculate and print the sum of digits of a user-entered integer.

Works with negative numbers by ignoring the sign character.

Works with negative numbers by ignoring the sign.

Definition at line 101 of file main.cpp.

```
00101                         {
00102     int number;
00103     std::cout « "Enter a number: ";
00104     if (!(std::cin » number)) { std::cin.clear();
      std::cin.ignore(std::numeric_limits<std::streamsize>::max(), '\n'); return; }
00105     std::string s = std::to_string(number);
00106     int sum = 0;
00107     for (char ch : s) {
00108         if (std::isdigit(static_cast<unsigned char>(ch))) sum += (ch - '0');
00109     }
00110     std::cout « "Sum of digits = " « sum « '\n';
00111 }
```

Here is the caller graph for this function:

4.4 main.h

Go to the documentation of this file.

```
00001 #ifndef TASKS_MENU_H
00002 #define TASKS_MENU_H
00003
00012
00013 #include <string>
00014
00015 #ifdef __cplusplus
00016 extern "C" {
00017 #endif
00018
00024 void task_ascii_table();
00025
00030 void task_max_of_three();
00031
00036 void task_right_triangle();
00037
00043 void task_check_vowel();
00044
00049 void task_multiplication_table();
00050
00055 void task_sum_digits();
00056
00062 void task_decimal_binary();
00063
00064 #ifdef __cplusplus
00065 }
00066 #endif
00067
00068 #endif // TASKS_MENU_H
```