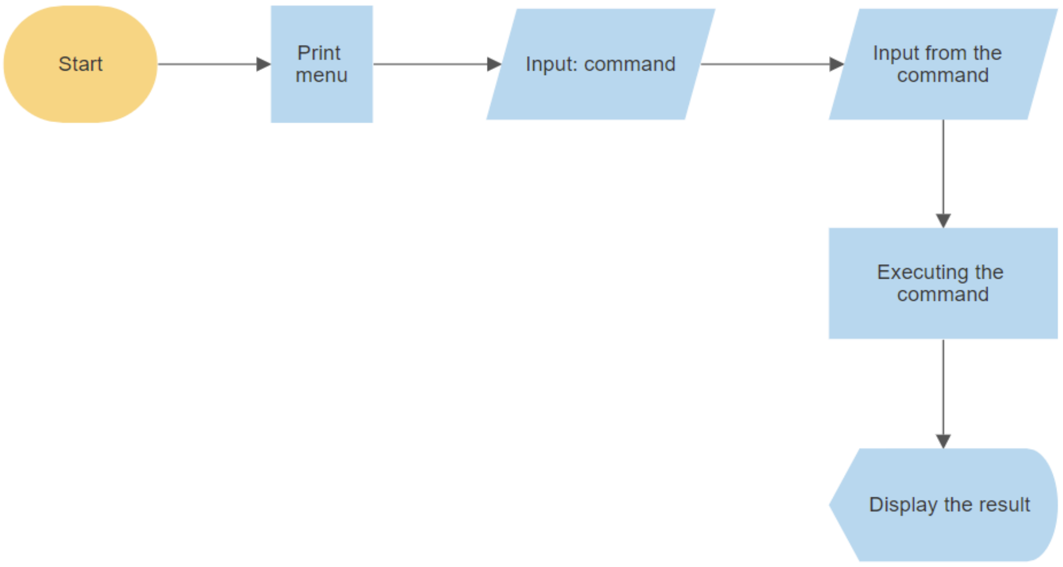
Author: Lupu Eduard

Base Calculator/ Converter

F1	Add two numbers in a specified base
F2	Subtract two numbers in a specified base
F3	Multiply two numbers in a specified base
F4	Divide a number by a digit in a specified base
F5	Conversion using successive division method
F6	Conversion using substitution method
F7	Conversion using rapid conversions method
F8	Conversing using intermediate base 10

Sub-algorithm’s diagram



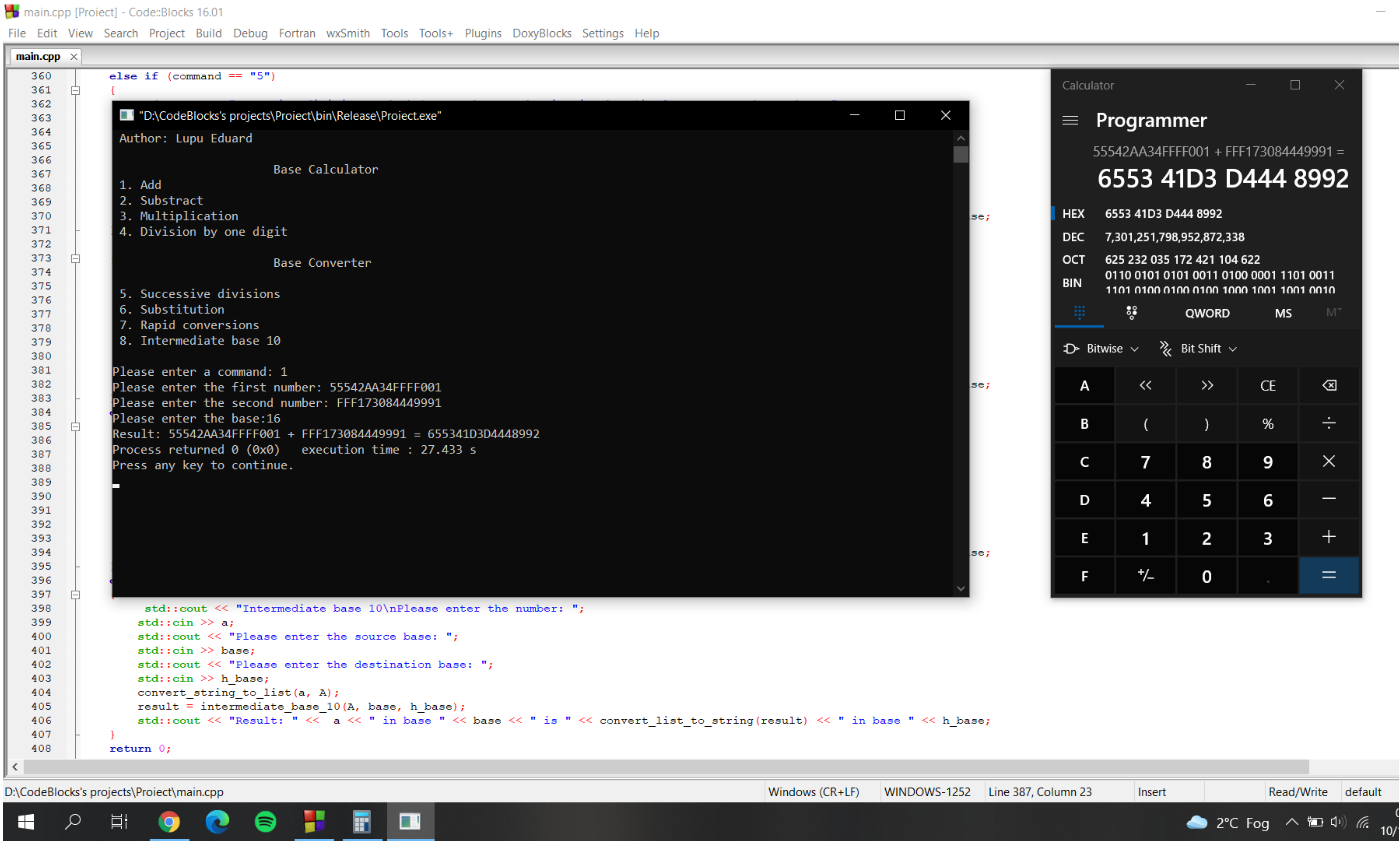
Author: Lupu Eduard

Used data type specification

string	To store the input of the number from the user
vector <int>	To store the digits of the number
int	To store the base value and other variables

Author: Lupu Eduard

Tests

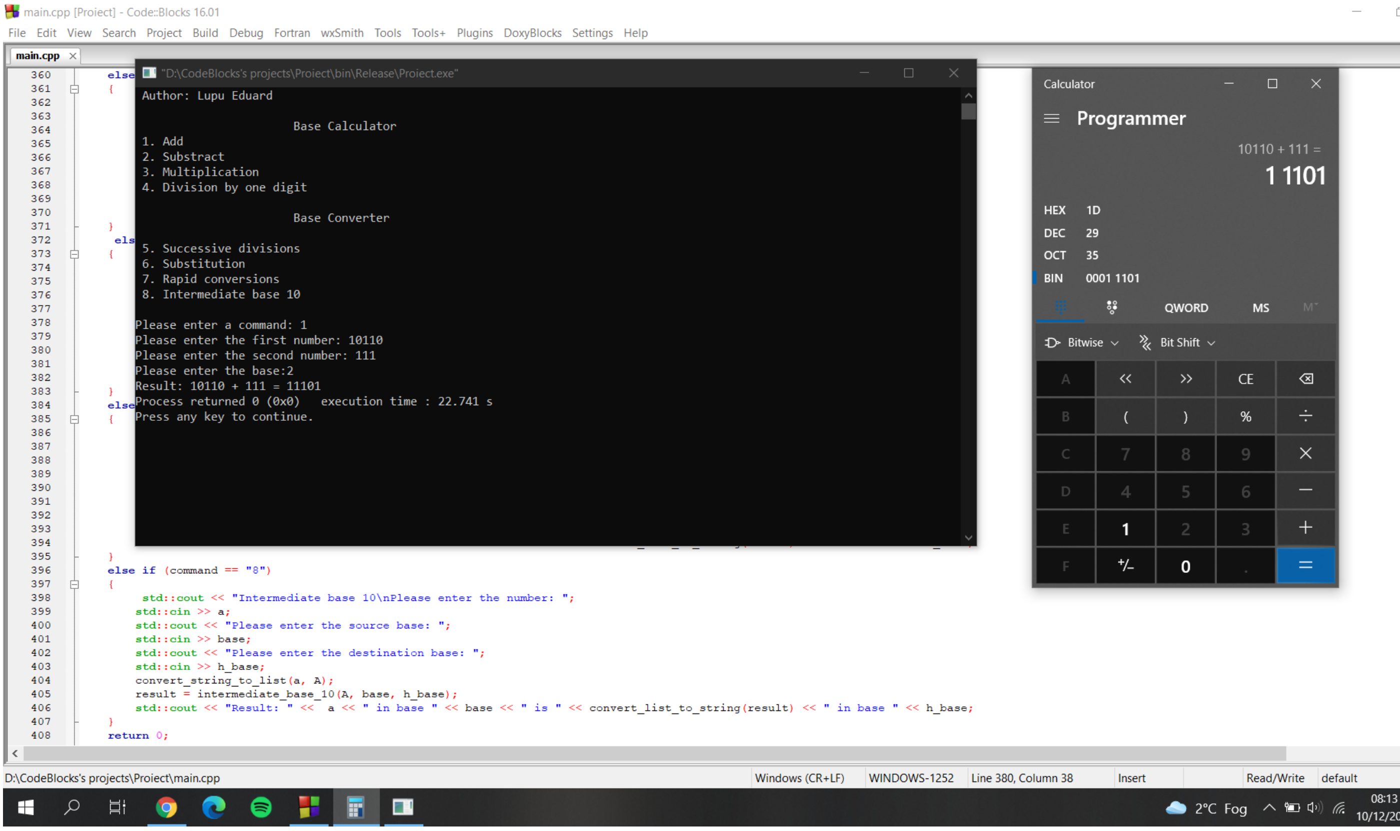


main.cpp [Proiect] - Code::Blocks 16.01
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
main.cpp
"D:\CodeBlocks's projects\Proiect\bin\Release\Proiect.exe"
Calculator
Programmer
55542AA34FFFF001 + FFF173084449991 =
6553 41D3 D444 8992
HEX 6553 41D3 D444 8992
DEC 7,301,251,798,952,872,338
OCT 625 232 035 172 421 104 622
BIN 0110 0101 0101 0011 0100 0001 1101 0011 1101 0100 0100 0100 1000 1001 1001 0010
QWORD
MS
Bitwise
Bit Shift
D:\CodeBlocks's projects\Proiect\main.cpp
Windows (CR+LF)
WINDOWS-1252
Line 387, Column 23
Insert
Read/Write
default
2°C Fog

```
    else if (command == "5")
    {
```

```
Author: Lupu Eduard

                        Base Calculator
1. Add
2. Substract
3. Multiplication
4. Division by one digit

                        Base Converter

5. Successive divisions
6. Substitution
7. Rapid conversions
8. Intermediate base 10

Please enter a command: 1
Please enter the first number: 55542AA34FFFF001
Please enter the second number: FFF173084449991
Please enter the base:16
Result: 55542AA34FFFF001 + FFF173084449991 = 655341D3D4448992
Process returned 0 (0x0)   execution time : 27.433 s
Press any key to continue.
```

```
         std::cout << "Intermediate base 10\nPlease enter the number: ";
        std::cin >> a;
        std::cout << "Please enter the source base: ";
        std::cin >> base;
        std::cout << "Please enter the destination base: ";
        std::cin >> h_base;
        convert_string_to_list(a, A);
        result = intermediate_base_10(A, base, h_base);
        std::cout << "Result: " <<  a << " in base " << base << " is " << convert_list_to_string(result) << " in base " << h_base;
    }
    return 0;
```

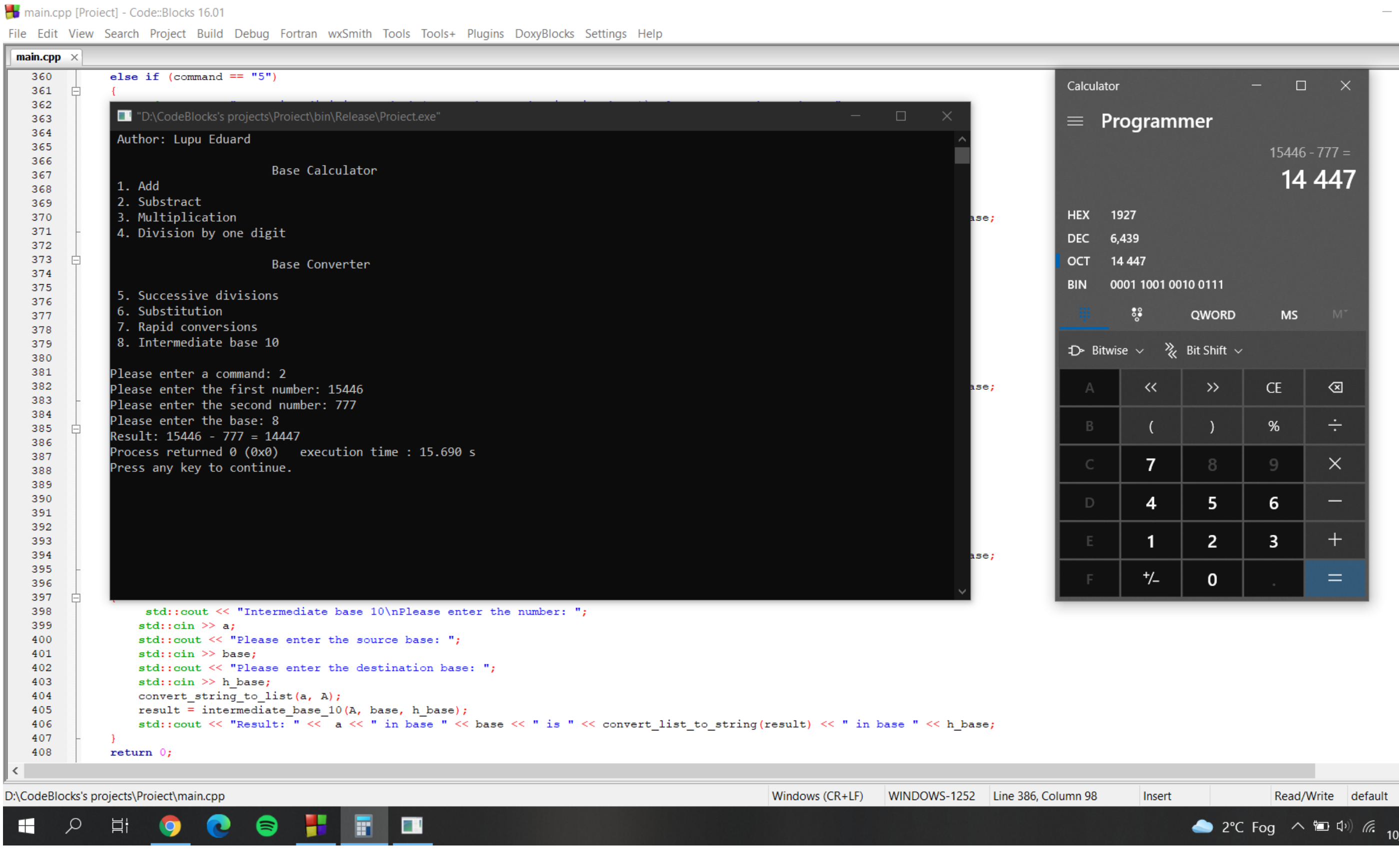


main.cpp [Proiect] - Code::Blocks 16.01
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
main.cpp
"D:\CodeBlocks's projects\Proiect\bin\Release\Proiect.exe"
Calculator
Programmer
10110 + 111 =
1 1101
HEX 1D
DEC 29
OCT 35
BIN 0001 1101
QWORD
MS
Bitwise
Bit Shift
D:\CodeBlocks's projects\Proiect\main.cpp
Windows (CR+LF)
WINDOWS-1252
Line 380, Column 38
Insert
Read/Write
default
2°C Fog
08:13
10/12/20

```
Author: Lupu Eduard

                        Base Calculator
1. Add
2. Substract
3. Multiplication
4. Division by one digit

                        Base Converter

5. Successive divisions
6. Substitution
7. Rapid conversions
8. Intermediate base 10

Please enter a command: 1
Please enter the first number: 10110
Please enter the second number: 111
Please enter the base:2
Result: 10110 + 111 = 11101
Process returned 0 (0x0)   execution time : 22.741 s
Press any key to continue.
```

```
    }
    else if (command == "8")
    {
         std::cout << "Intermediate base 10\nPlease enter the number: ";
        std::cin >> a;
        std::cout << "Please enter the source base: ";
        std::cin >> base;
        std::cout << "Please enter the destination base: ";
        std::cin >> h_base;
        convert_string_to_list(a, A);
        result = intermediate_base_10(A, base, h_base);
        std::cout << "Result: " <<  a << " in base " << base << " is " << convert_list_to_string(result) << " in base " << h_base;
    }
    return 0;
```

Author: Lupu Eduard



main.cpp [Proiect] - Code::Blocks 16.01
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
main.cpp

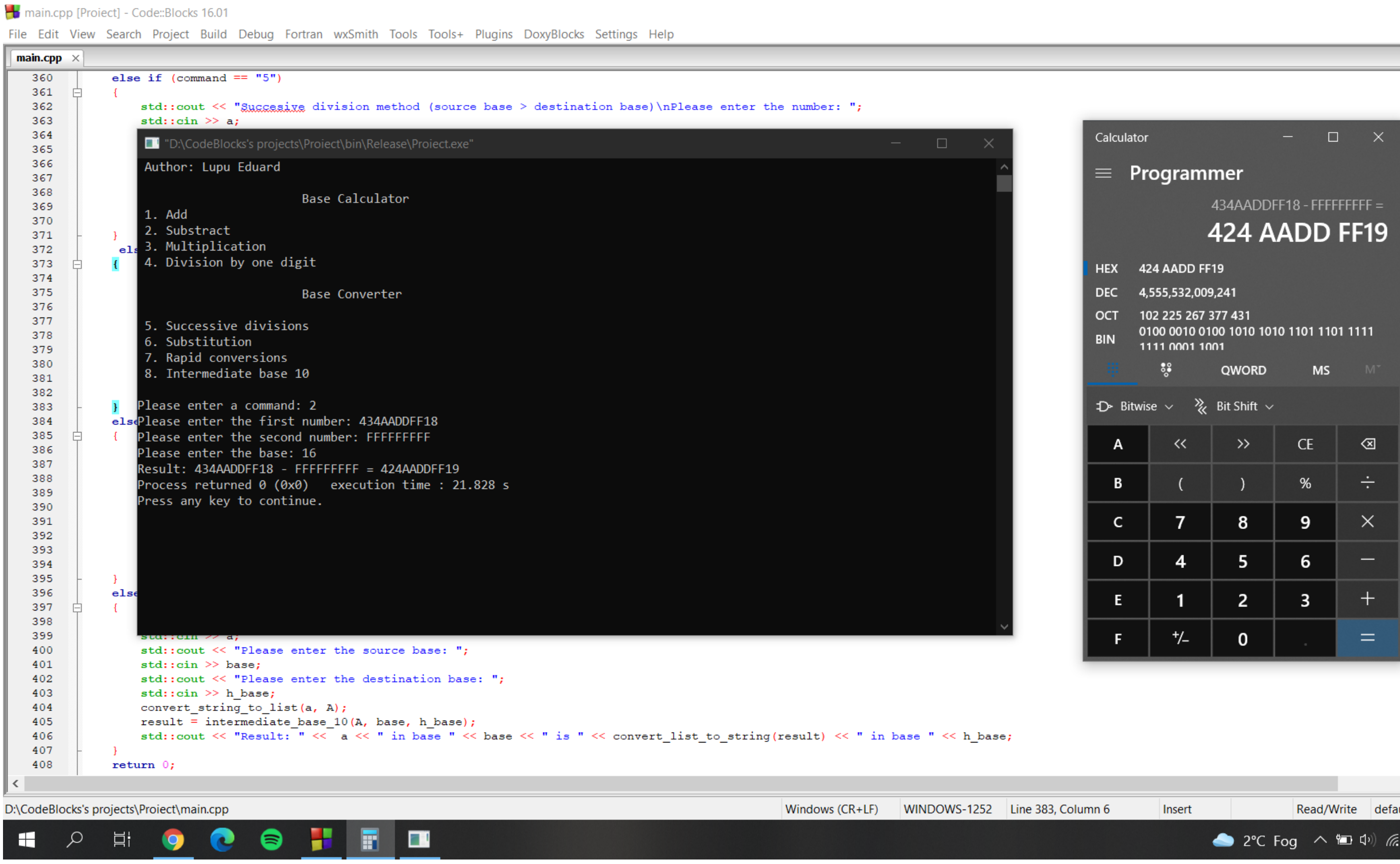
```cpp
else if (command == "5")
{
```

"D:\CodeBlocks's projects\Proiect\bin\Release\Proiect.exe"

```
Author: Lupu Eduard

                      Base Calculator
1. Add
2. Substract
3. Multiplication
4. Division by one digit

                      Base Converter

5. Successive divisions
6. Substitution
7. Rapid conversions
8. Intermediate base 10

Please enter a command: 2
Please enter the first number: 15446
Please enter the second number: 777
Please enter the base: 8
Result: 15446 - 777 = 14447
Process returned 0 (0x0)   execution time : 15.690 s
Press any key to continue.
```

```cpp
     std::cout << "Intermediate base 10\nPlease enter the number: ";
    std::cin >> a;
    std::cout << "Please enter the source base: ";
    std::cin >> base;
    std::cout << "Please enter the destination base: ";
    std::cin >> h_base;
    convert_string_to_list(a, A);
    result = intermediate_base_10(A, base, h_base);
    std::cout << "Result: " <<  a << " in base " << base << " is " << convert_list_to_string(result) << " in base " << h_base;
}
return 0;
```

Calculator
Programmer
15446 - 777 =
14 447
HEX 1927
DEC 6,439
OCT 14 447
BIN 0001 1001 0010 0111
QWORD
MS
Bitwise
Bit Shift
D:\CodeBlocks's projects\Proiect\main.cpp
Windows (CR+LF)
WINDOWS-1252
Line 386, Column 98
Insert
Read/Write
default
2°C Fog



main.cpp [Proiect] - Code::Blocks 16.01
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
main.cpp

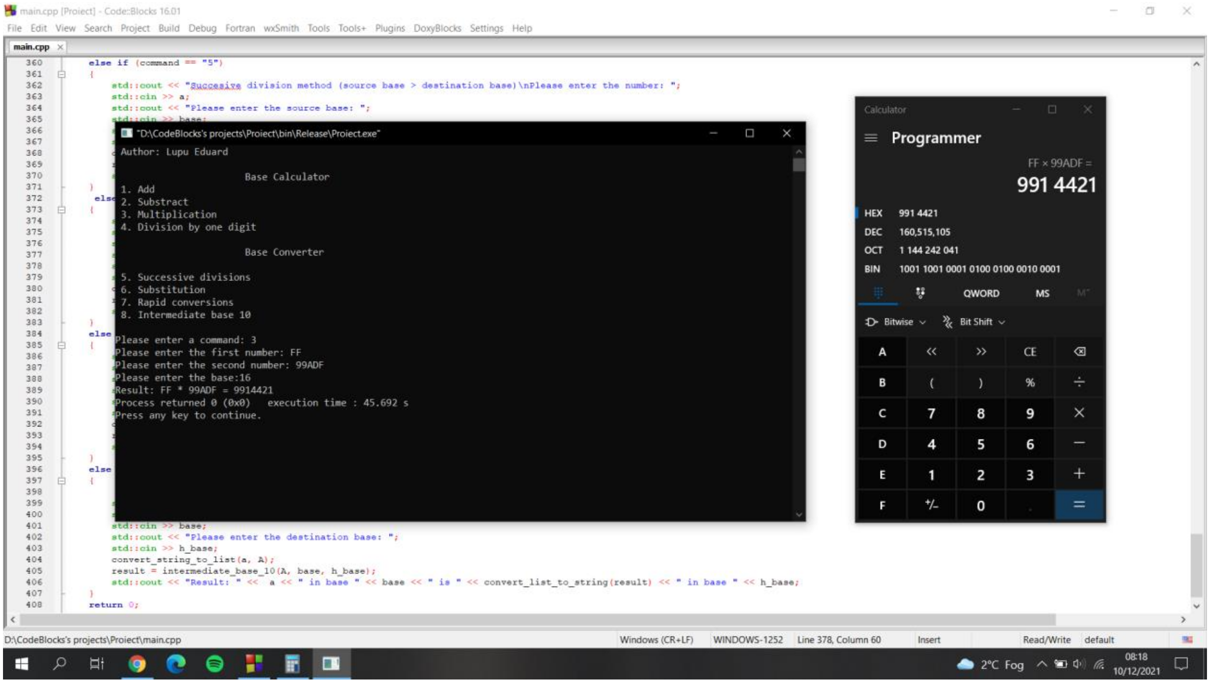
```cpp
else if (command == "5")
{
    std::cout << "Succesive division method (source base > destination base)\nPlease enter the number: ";
    std::cin >> a;
```

"D:\CodeBlocks's projects\Proiect\bin\Release\Proiect.exe"

```
Author: Lupu Eduard

                      Base Calculator

1. Add
2. Substract
3. Multiplication
4. Division by one digit

                      Base Converter

5. Successive divisions
6. Substitution
7. Rapid conversions
8. Intermediate base 10

Please enter a command: 2
Please enter the first number: 434AADDFF18
Please enter the second number: FFFFFFFFF
Please enter the base: 16
Result: 434AADDFF18 - FFFFFFFFF = 424AADDFF19
Process returned 0 (0x0)   execution time : 21.828 s
Press any key to continue.
```

```cpp
    std::cout << "Please enter the source base: ";
    std::cin >> base;
    std::cout << "Please enter the destination base: ";
    std::cin >> h_base;
    convert_string_to_list(a, A);
    result = intermediate_base_10(A, base, h_base);
    std::cout << "Result: " <<  a << " in base " << base << " is " << convert_list_to_string(result) << " in base " << h_base;
}
return 0;
```

Author: Lupu Eduard



main.cpp [Proiect] - Code::Blocks 16.01
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
main.cpp

```cpp
    else if (command == "5")
    {
        std::cout << "Succesive division method (source base > destination base)\nPlease enter the number: ";
        std::cin >> a;
        std::cout << "Please enter the source base: ";
        std::cin >> base;
        ...
        std::cin >> base;
        std::cout << "Please enter the destination base: ";
        std::cin >> h_base;
        convert_string_to_list(a, A);
        result = intermediate_base_10(A, base, h_base);
        std::cout << "Result: " <<  a << " in base " << base << " is " << convert_list_to_string(result) << " in base " << h_base;
    }
    return 0;
```

"D:\CodeBlocks's projects\Proiect\bin\Release\Proiect.exe"

```
Author: Lupu Eduard

                        Base Calculator
1. Add
2. Substract
3. Multiplication
4. Division by one digit

                        Base Converter

5. Successive divisions
6. Substitution
7. Rapid conversions
8. Intermediate base 10

Please enter a command: 3
Please enter the first number: FF
Please enter the second number: 99ADF
Please enter the base:16
Result: FF * 99ADF = 9914421
Process returned 0 (0x0)   execution time : 45.692 s
Press any key to continue.
```

Calculator

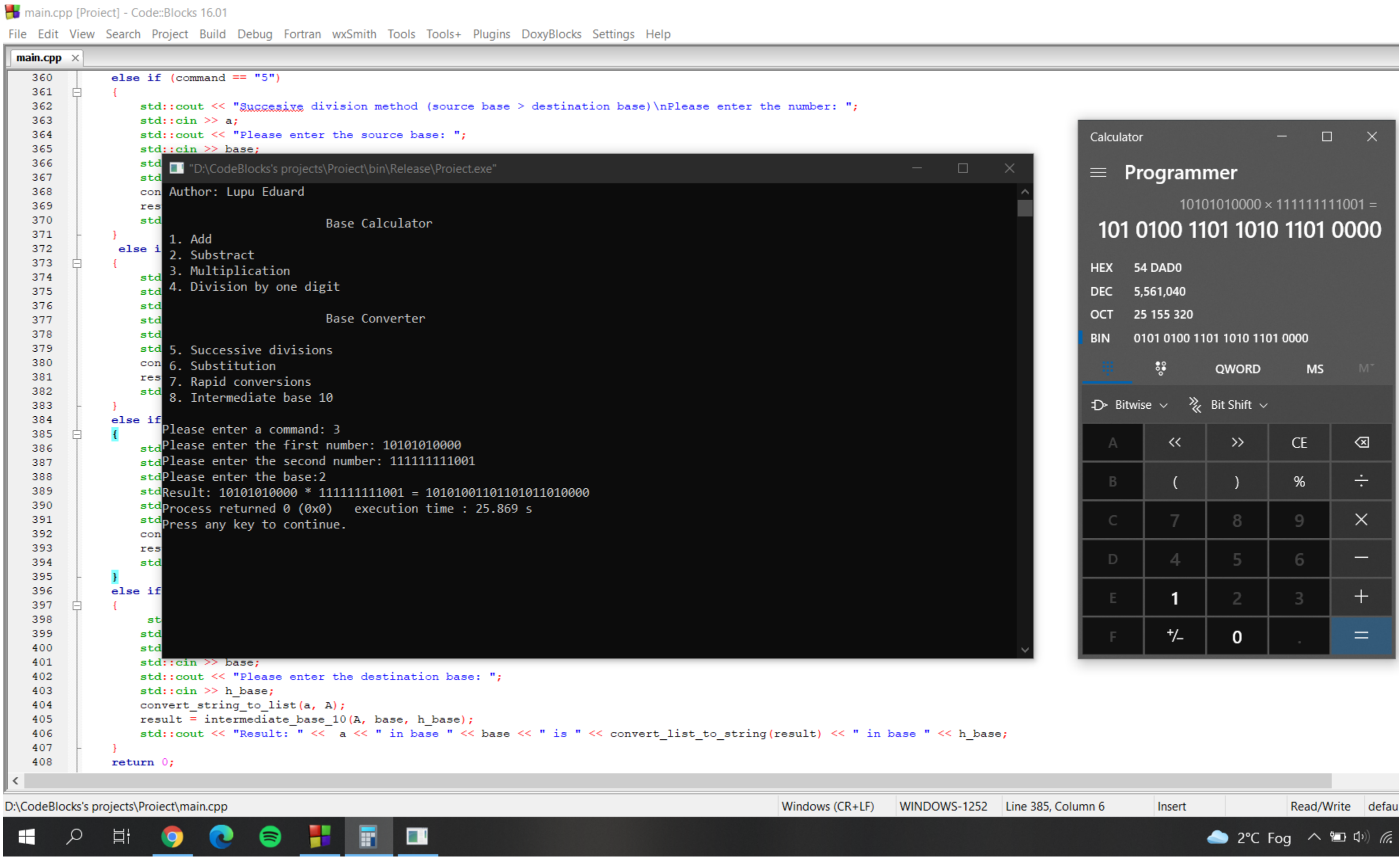
Programmer

FF × 99ADF =

991 4421

HEX 991 4421
DEC 160,515,105
OCT 1 144 242 041
BIN 1001 1001 0001 0100 0100 0010 0001

QWORD
MS
Bitwise
Bit Shift
A « » CE ⌫
B ( ) % ÷
C 7 8 9 ×
D 4 5 6 −
E 1 2 3 +
F +/- 0 . =
D:\CodeBlocks's projects\Proiect\main.cpp
Windows (CR+LF)
WINDOWS-1252
Line 378, Column 60
Insert
Read/Write
default
2°C Fog
08:18
10/12/2021



main.cpp [Proiect] - Code::Blocks 16.01
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
main.cpp

```cpp
    else if (command == "5")
    {
        std::cout << "Succesive division method (source base > destination base)\nPlease enter the number: ";
        std::cin >> a;
        std::cout << "Please enter the source base: ";
        std::cin >> base;
        ...
        std::cin >> base;
        std::cout << "Please enter the destination base: ";
        std::cin >> h_base;
        convert_string_to_list(a, A);
        result = intermediate_base_10(A, base, h_base);
        std::cout << "Result: " <<  a << " in base " << base << " is " << convert_list_to_string(result) << " in base " << h_base;
    }
    return 0;
```

"D:\CodeBlocks's projects\Proiect\bin\Release\Proiect.exe"

```
Author: Lupu Eduard

                        Base Calculator
1. Add
2. Substract
3. Multiplication
4. Division by one digit

                        Base Converter

5. Successive divisions
6. Substitution
7. Rapid conversions
8. Intermediate base 10

Please enter a command: 3
Please enter the first number: 10101010000
Please enter the second number: 111111111001
Please enter the base:2
Result: 10101010000 * 111111111001 = 10101001101101011010000
Process returned 0 (0x0)   execution time : 25.869 s
Press any key to continue.
```

Calculator

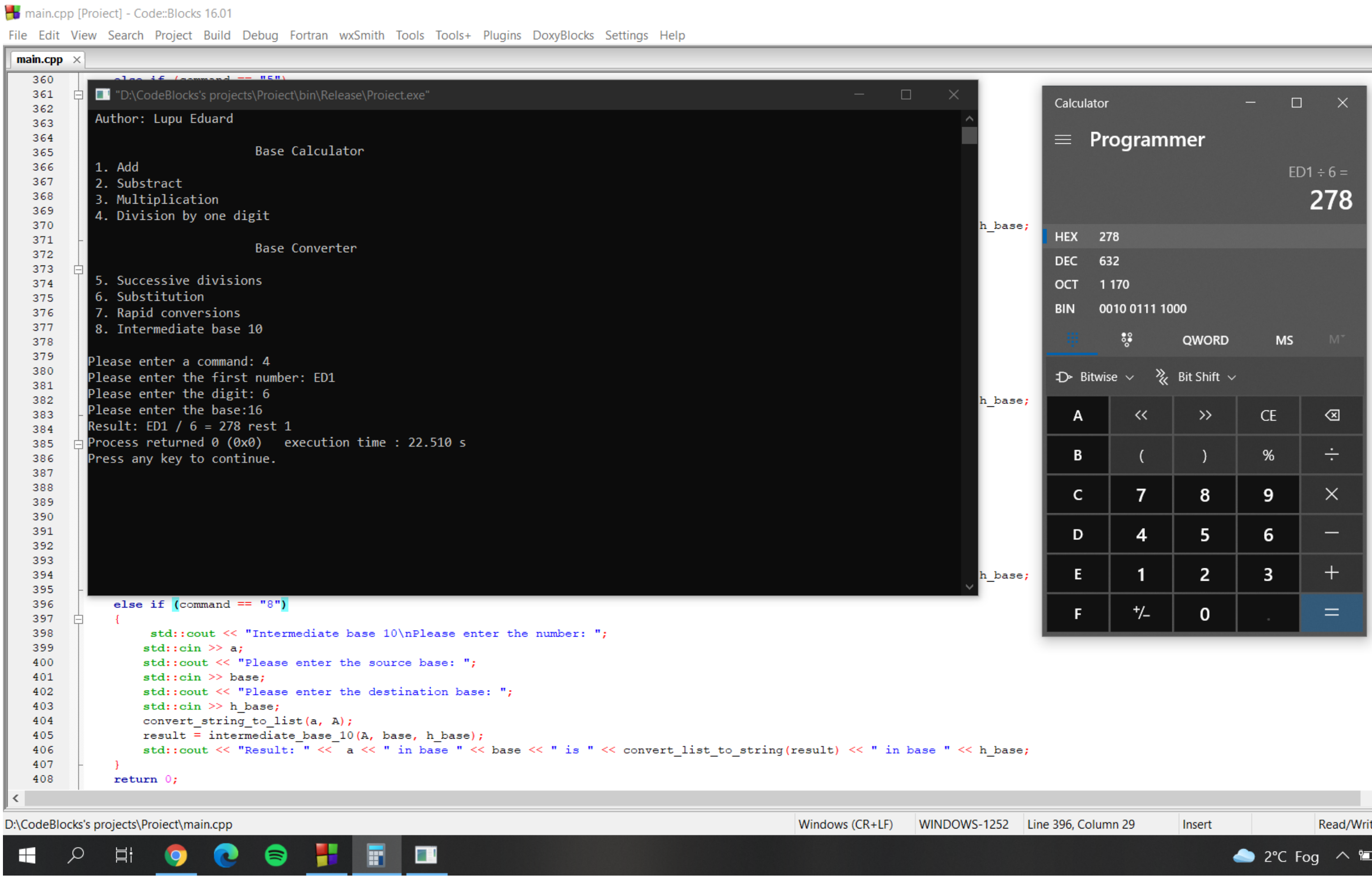
Programmer

10101010000 × 111111111001 =

101 0100 1101 1010 1101 0000

HEX 54 DAD0
DEC 5,561,040
OCT 25 155 320
BIN 0101 0100 1101 1010 1101 0000

Author: Lupu Eduard


main.cpp [Proiect] - Code::Blocks 16.01
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
main.cpp
"D:\CodeBlocks's projects\Proiect\bin\Release\Proiect.exe"
Calculator
Programmer
QWORD MS
Bitwise Bit Shift

```
Author: Lupu Eduard

                      Base Calculator
1. Add
2. Substract
3. Multiplication
4. Division by one digit

                      Base Converter

5. Successive divisions
6. Substitution
7. Rapid conversions
8. Intermediate base 10

Please enter a command: 4
Please enter the first number: ED1
Please enter the digit: 6
Please enter the base:16
Result: ED1 / 6 = 278 rest 1
Process returned 0 (0x0)   execution time : 22.510 s
Press any key to continue.
```

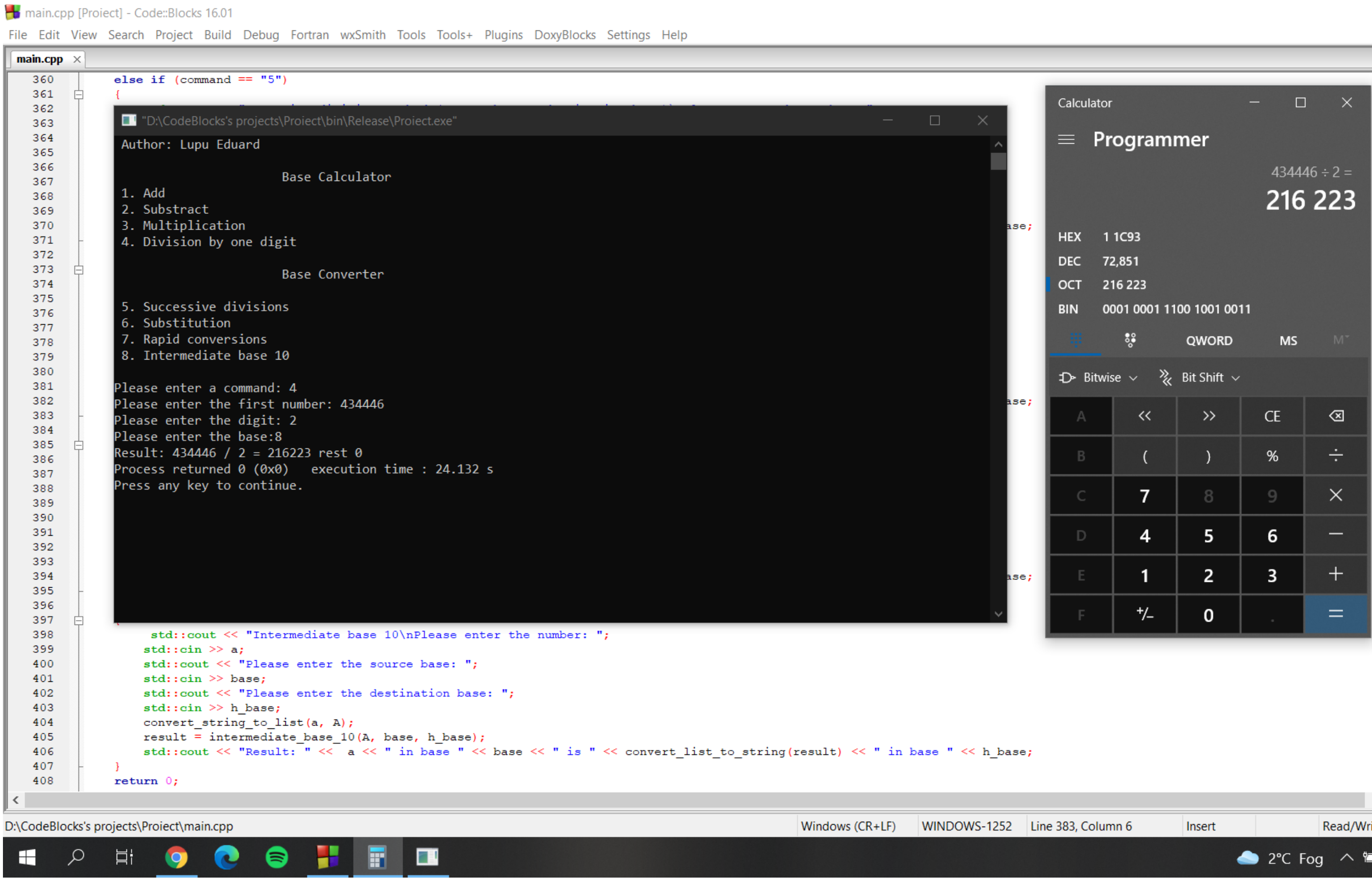
```
        else if (command == "8")
        {
             std::cout << "Intermediate base 10\nPlease enter the number: ";
            std::cin >> a;
            std::cout << "Please enter the source base: ";
            std::cin >> base;
            std::cout << "Please enter the destination base: ";
            std::cin >> h_base;
            convert_string_to_list(a, A);
            result = intermediate_base_10(A, base, h_base);
            std::cout << "Result: " <<  a << " in base " << base << " is " << convert_list_to_string(result) << " in base " << h_base;
        }
        return 0;
```

ED1 ÷ 6 =
278

HEX 278
DEC 632
OCT 1 170
BIN 0010 0111 1000

D:\CodeBlocks's projects\Proiect\main.cpp | Windows (CR+LF) | WINDOWS-1252 | Line 396, Column 29 | Insert | Read/Wri

2°C Fog


main.cpp [Proiect] - Code::Blocks 16.01
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
main.cpp
"D:\CodeBlocks's projects\Proiect\bin\Release\Proiect.exe"
Calculator
Programmer
QWORD MS
Bitwise Bit Shift

```
        else if (command == "5")
        {
```

```
Author: Lupu Eduard

                      Base Calculator
1. Add
2. Substract
3. Multiplication
4. Division by one digit

                      Base Converter

5. Successive divisions
6. Substitution
7. Rapid conversions
8. Intermediate base 10

Please enter a command: 4
Please enter the first number: 434446
Please enter the digit: 2
Please enter the base:8
Result: 434446 / 2 = 216223 rest 0
Process returned 0 (0x0)   execution time : 24.132 s
Press any key to continue.
```

```
             std::cout << "Intermediate base 10\nPlease enter the number: ";
            std::cin >> a;
            std::cout << "Please enter the source base: ";
            std::cin >> base;
            std::cout << "Please enter the destination base: ";
            std::cin >> h_base;
            convert_string_to_list(a, A);
            result = intermediate_base_10(A, base, h_base);
            std::cout << "Result: " <<  a << " in base " << base << " is " << convert_list_to_string(result) << " in base " << h_base;
        }
        return 0;
```

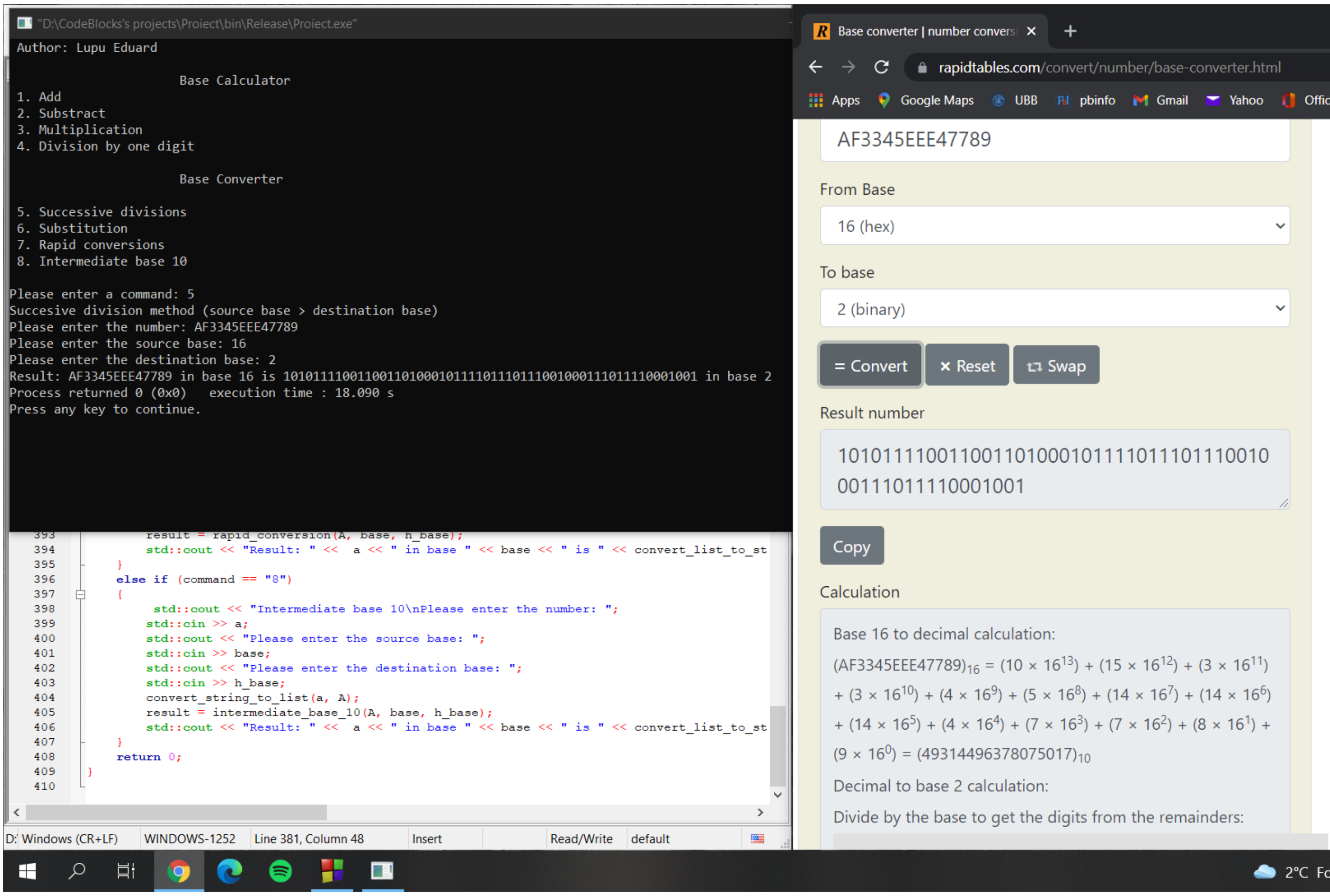
434446 ÷ 2 =
216 223

HEX 1 1C93
DEC 72,851
OCT 216 223
BIN 0001 0001 1100 1001 0011

D:\CodeBlocks's projects\Proiect\main.cpp | Windows (CR+LF) | WINDOWS-1252 | Line 383, Column 6 | Insert | Read/Wri

2°C Fog

Author: Lupu Eduard



"D:\CodeBlocks's projects\Proiect\bin\Release\Proiect.exe"

```
Author: Lupu Eduard

                      Base Calculator
 1. Add
 2. Substract
 3. Multiplication
 4. Division by one digit

                      Base Converter

 5. Successive divisions
 6. Substitution
 7. Rapid conversions
 8. Intermediate base 10

Please enter a command: 5
Succesive division method (source base > destination base)
Please enter the number: AF3345EEE47789
Please enter the source base: 16
Please enter the destination base: 2
Result: AF3345EEE47789 in base 16 is 10101111001100110100010111101110111001000111011110001001 in base 2
Process returned 0 (0x0)   execution time : 18.090 s
Press any key to continue.
```

```cpp
            result = rapid_conversion(A, base, h_base);
            std::cout << "Result: " <<  a << " in base " << base << " is " << convert_list_to_st
        }
        else if (command == "8")
        {
             std::cout << "Intermediate base 10\nPlease enter the number: ";
            std::cin >> a;
            std::cout << "Please enter the source base: ";
            std::cin >> base;
            std::cout << "Please enter the destination base: ";
            std::cin >> h_base;
            convert_string_to_list(a, A);
            result = intermediate_base_10(A, base, h_base);
            std::cout << "Result: " <<  a << " in base " << base << " is " << convert_list_to_st
        }
        return 0;
}
```

AF3345EEE47789

From Base

16 (hex)

To base

2 (binary)

Result number

10101111001100110100010111101110111001000111011110001001

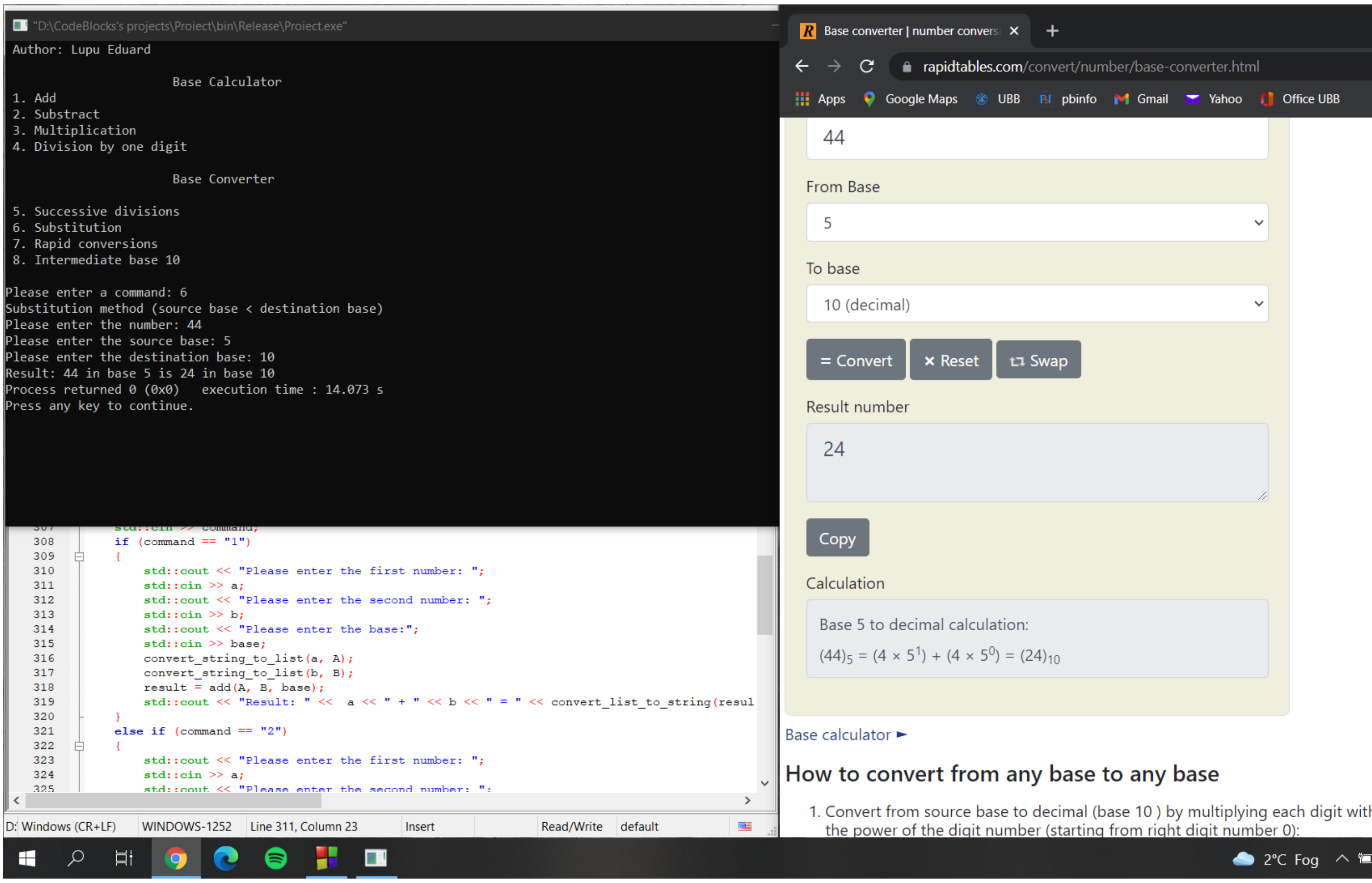
Calculation

Base 16 to decimal calculation:

$(AF3345EEE47789)_{16} = (10 \times 16^{13}) + (15 \times 16^{12}) + (3 \times 16^{11}) + (3 \times 16^{10}) + (4 \times 16^{9}) + (5 \times 16^{8}) + (14 \times 16^{7}) + (14 \times 16^{6}) + (14 \times 16^{5}) + (4 \times 16^{4}) + (7 \times 16^{3}) + (7 \times 16^{2}) + (8 \times 16^{1}) + (9 \times 16^{0}) = (49314496378075017)_{10}$

Decimal to base 2 calculation:

Divide by the base to get the digits from the remainders:



"D:\CodeBlocks's projects\Proiect\bin\Release\Proiect.exe"

```
Author: Lupu Eduard

                      Base Calculator
 1. Add
 2. Substract
 3. Multiplication
 4. Division by one digit

                      Base Converter

 5. Successive divisions
 6. Substitution
 7. Rapid conversions
 8. Intermediate base 10

Please enter a command: 6
Substitution method (source base < destination base)
Please enter the number: 44
Please enter the source base: 5
Please enter the destination base: 10
Result: 44 in base 5 is 24 in base 10
Process returned 0 (0x0)   execution time : 14.073 s
Press any key to continue.
```

```cpp
        std::cin >> command;
        if (command == "1")
        {
            std::cout << "Please enter the first number: ";
            std::cin >> a;
            std::cout << "Please enter the second number: ";
            std::cin >> b;
            std::cout << "Please enter the base:";
            std::cin >> base;
            convert_string_to_list(a, A);
            convert_string_to_list(b, B);
            result = add(A, B, base);
            std::cout << "Result: " <<  a << " + " << b << " = " << convert_list_to_string(resul
        }
        else if (command == "2")
        {
            std::cout << "Please enter the first number: ";
            std::cin >> a;
            std::cout << "Please enter the second number: ";
```

44

From Base

5

To base

10 (decimal)

Result number

24

Calculation

Base 5 to decimal calculation:

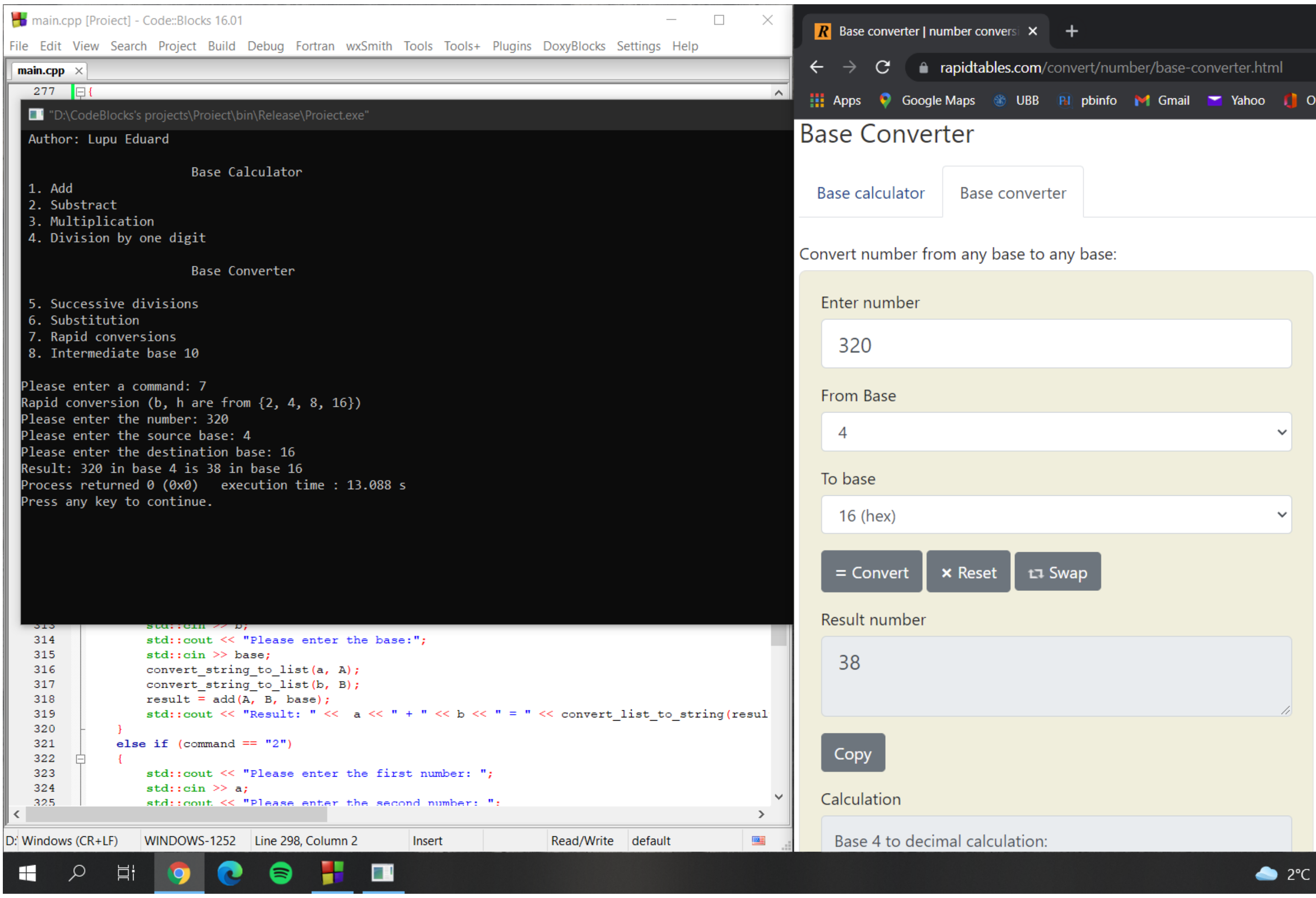
$(44)_5 = (4 \times 5^1) + (4 \times 5^0) = (24)_{10}$

Base calculator ►

How to convert from any base to any base

1. Convert from source base to decimal (base 10) by multiplying each digit with the power of the digit number (starting from right digit number 0):

Author: Lupu Eduard



main.cpp [Proiect] - Code::Blocks 16.01
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
main.cpp
"D:\CodeBlocks's projects\Proiect\bin\Release\Proiect.exe"

```
Author: Lupu Eduard

                        Base Calculator
 1. Add
 2. Substract
 3. Multiplication
 4. Division by one digit

                        Base Converter

 5. Successive divisions
 6. Substitution
 7. Rapid conversions
 8. Intermediate base 10

Please enter a command: 7
Rapid conversion (b, h are from {2, 4, 8, 16})
Please enter the number: 320
Please enter the source base: 4
Please enter the destination base: 16
Result: 320 in base 4 is 38 in base 16
Process returned 0 (0x0)   execution time : 13.088 s
Press any key to continue.
```

```
            std::cin >> b;
            std::cout << "Please enter the base:";
            std::cin >> base;
            convert_string_to_list(a, A);
            convert_string_to_list(b, B);
            result = add(A, B, base);
            std::cout << "Result: " <<  a << " + " << b << " = " << convert_list_to_string(resul
        }
        else if (command == "2")
        {
            std::cout << "Please enter the first number: ";
            std::cin >> a;
            std::cout << "Please enter the second number: ";
```

Base Converter

Base calculator | Base converter

Convert number from any base to any base:

Enter number: 320

From Base: 4

To base: 16 (hex)

Result number: 38

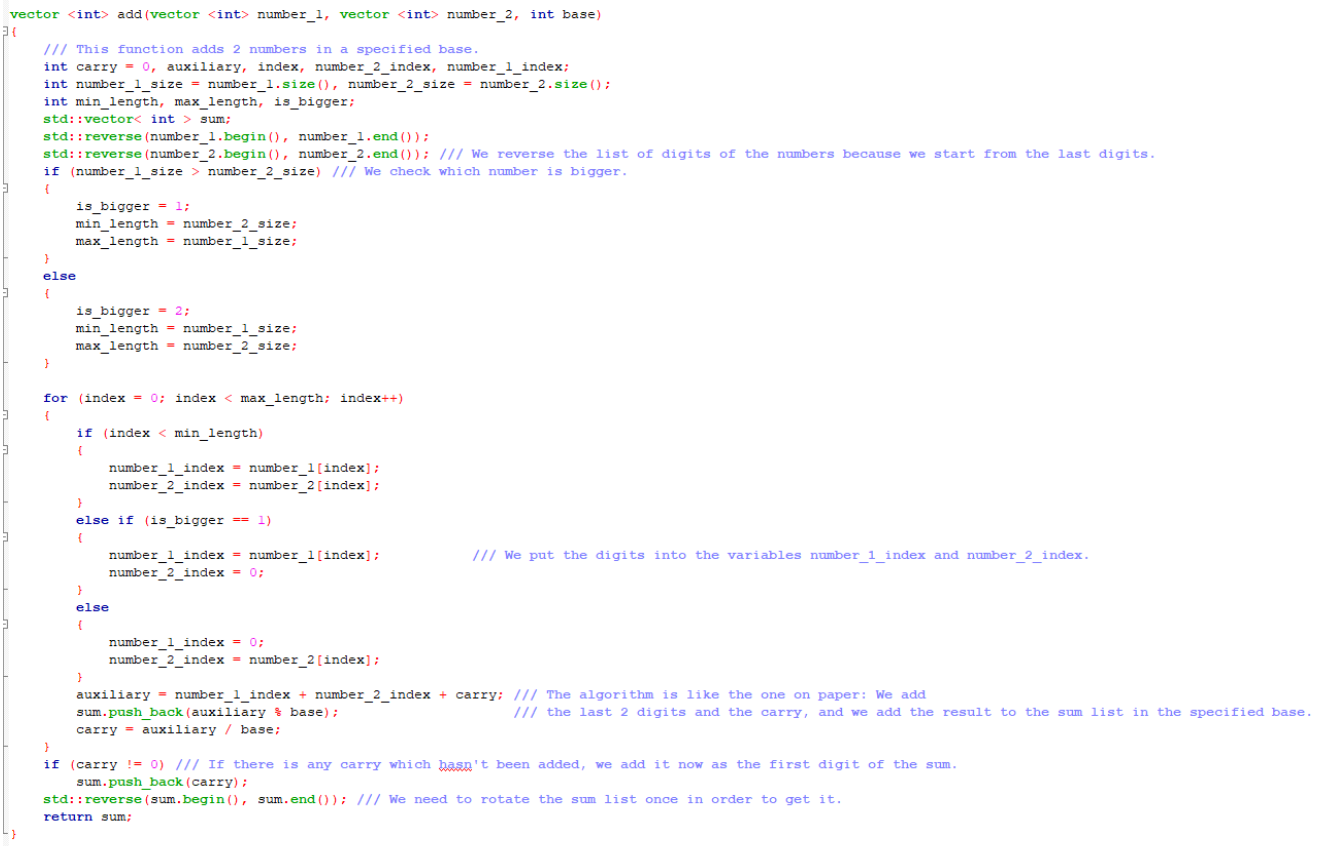
Calculation

Base 4 to decimal calculation:

Author: Lupu Eduard

Algorithms

Addition

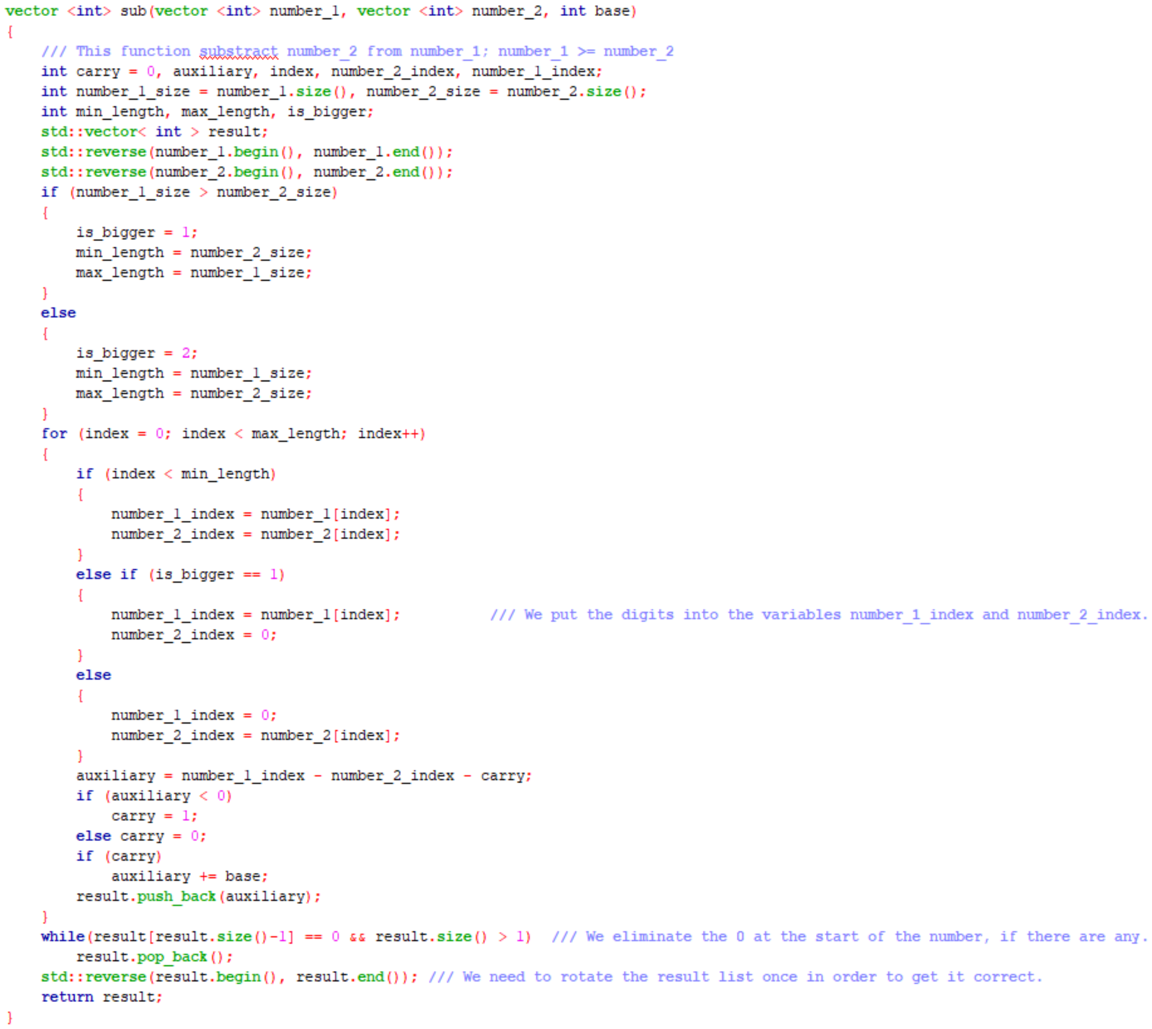


```
vector <int> add(vector <int> number_1, vector <int> number_2, int base)
{
    /// This function adds 2 numbers in a specified base.
    int carry = 0, auxiliary, index, number_2_index, number_1_index;
    int number_1_size = number_1.size(), number_2_size = number_2.size();
    int min_length, max_length, is_bigger;
    std::vector< int > sum;
    std::reverse(number_1.begin(), number_1.end());
    std::reverse(number_2.begin(), number_2.end()); /// We reverse the list of digits of the numbers because we start from the last digits.
    if (number_1_size > number_2_size) /// We check which number is bigger.
    {
        is_bigger = 1;
        min_length = number_2_size;
        max_length = number_1_size;
    }
    else
    {
        is_bigger = 2;
        min_length = number_1_size;
        max_length = number_2_size;
    }

    for (index = 0; index < max_length; index++)
    {
        if (index < min_length)
        {
            number_1_index = number_1[index];
            number_2_index = number_2[index];
        }
        else if (is_bigger == 1)
        {
            number_1_index = number_1[index];           /// We put the digits into the variables number_1_index and number_2_index.
            number_2_index = 0;
        }
        else
        {
            number_1_index = 0;
            number_2_index = number_2[index];
        }
        auxiliary = number_1_index + number_2_index + carry; /// The algorithm is like the one on paper: We add
        sum.push_back(auxiliary % base);                     /// the last 2 digits and the carry, and we add the result to the sum list in the specified base.
        carry = auxiliary / base;
    }
    if (carry != 0) /// If there is any carry which hasn't been added, we add it now as the first digit of the sum.
        sum.push_back(carry);
    std::reverse(sum.begin(), sum.end()); /// We need to rotate the sum list once in order to get it.
    return sum;
}
```

- The addition algorithm is straightforward: we simulate the process of adding 2 numbers on paper.
- We take every 2 last digits of the numbers, we add them and add the carry, then we store the sum to the result vector list.
- Add the end, we add the last carry and return the sum of 2 numbers.

Author: Lupu Eduard

Subtraction



```
vector <int> sub(vector <int> number_1, vector <int> number_2, int base)
{
    /// This function substract number_2 from number_1; number_1 >= number_2
    int carry = 0, auxiliary, index, number_2_index, number_1_index;
    int number_1_size = number_1.size(), number_2_size = number_2.size();
    int min_length, max_length, is_bigger;
    std::vector< int > result;
    std::reverse(number_1.begin(), number_1.end());
    std::reverse(number_2.begin(), number_2.end());
    if (number_1_size > number_2_size)
    {
        is_bigger = 1;
        min_length = number_2_size;
        max_length = number_1_size;
    }
    else
    {
        is_bigger = 2;
        min_length = number_1_size;
        max_length = number_2_size;
    }
    for (index = 0; index < max_length; index++)
    {
        if (index < min_length)
        {
            number_1_index = number_1[index];
            number_2_index = number_2[index];
        }
        else if (is_bigger == 1)
        {
            number_1_index = number_1[index];          /// We put the digits into the variables number_1_index and number_2_index.
            number_2_index = 0;
        }
        else
        {
            number_1_index = 0;
            number_2_index = number_2[index];
        }
        auxiliary = number_1_index - number_2_index - carry;
        if (auxiliary < 0)
            carry = 1;
        else carry = 0;
        if (carry)
            auxiliary += base;
        result.push_back(auxiliary);
    }
    while(result[result.size()-1] == 0 && result.size() > 1)  /// We eliminate the 0 at the start of the number, if there are any.
        result.pop_back();
    std::reverse(result.begin(), result.end()); /// We need to rotate the result list once in order to get it correct.
    return result;
}
```

- The subtraction algorithm is similar to the addition. We simulate the process of subtracting 2 numbers on paper.
- We take the last 2 digits, subtract them and the carry, and if the result is negative, we actualize the carry, and if not, we reset the carry.
- This process may be leading to a result where you have 0 at the start. So in the, we delete those and then we return the result.

Author: Lupu Eduard

Multiplication

```cpp
vector <int> mul(vector <int> number_1, vector <int> number_2, int base)
{
    /// This function multiplies 2 numbers in a specified base.
    int carry = 0, index, index2;
    int number_1_size = number_1.size(), number_2_size = number_2.size();
    std::vector< int > result(100, 0);
    std::reverse(number_1.begin(), number_1.end());
    std::reverse(number_2.begin(), number_2.end());
    for (index = 0; index < number_1_size; index++)            /// We multiply every 2 digits of the numbers and store the result.
    {
        for (index2 = 0; index2 < number_2_size; index2++)
        {
            result[index + index2] += number_1[index] * number_2[index2];
        }
    }
    for (index = 0; index < (int)result.size(); index++)  /// We add the carries.
    {
        result[index] += carry;
        carry = result[index] / base;
        result[index] = result[index] % base;
    }
    while (carry)                    /// If the carry isn't 0, we have to add him at the start of the result until he is.
    {
        result.push_back(carry % base);
        carry = carry / base;
    }
    while(result[result.size()-1] == 0 && result.size() > 1) /// We eliminate 0 from the start, in case they are any
        result.pop_back();
    std::reverse(result.begin(), result.end());
    return result;
}
```

- We first multiply every 2 digit from the numbers.
- Then we add the carries.
- And if at the end there are still carries left, we add them to the start of the number.
- Because we initialized our result vector with 0, we need to remove them and then we return the result.

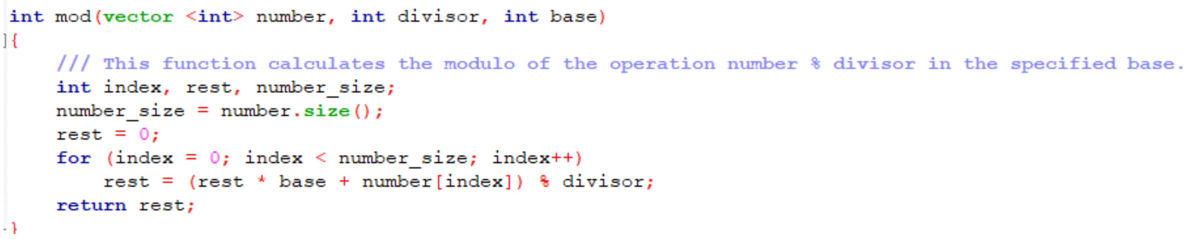
Division

```cpp
vector <int> div(vector <int> number, int divisor, int base)
{
    /// This function divides the number <number> in base <base> with the one digit divisor <divisor>
    int carry = 0, index;
    int number_size = number.size();
    for (index = 0; index < number_size; index++)
    {
        carry = base * carry + number[index];
        number[index] = carry / divisor;
        carry = carry % divisor;
    }
    std::reverse(number.begin(), number.end());
    while(number[number.size()-1] == 0 && number.size() > 1) /// We remove 0 from the start
        number.pop_back();
    std::reverse(number.begin(), number.end());
    return number; /// The result of number / divisor in the specified base.
}
```

The function returns the result of the operation number / divisor.

Author: Lupu Eduard

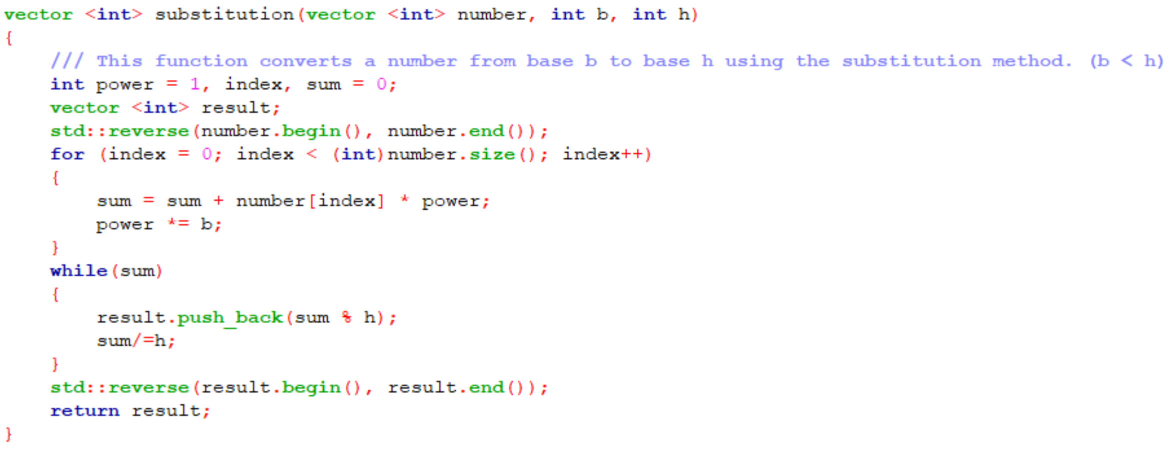
Modulo



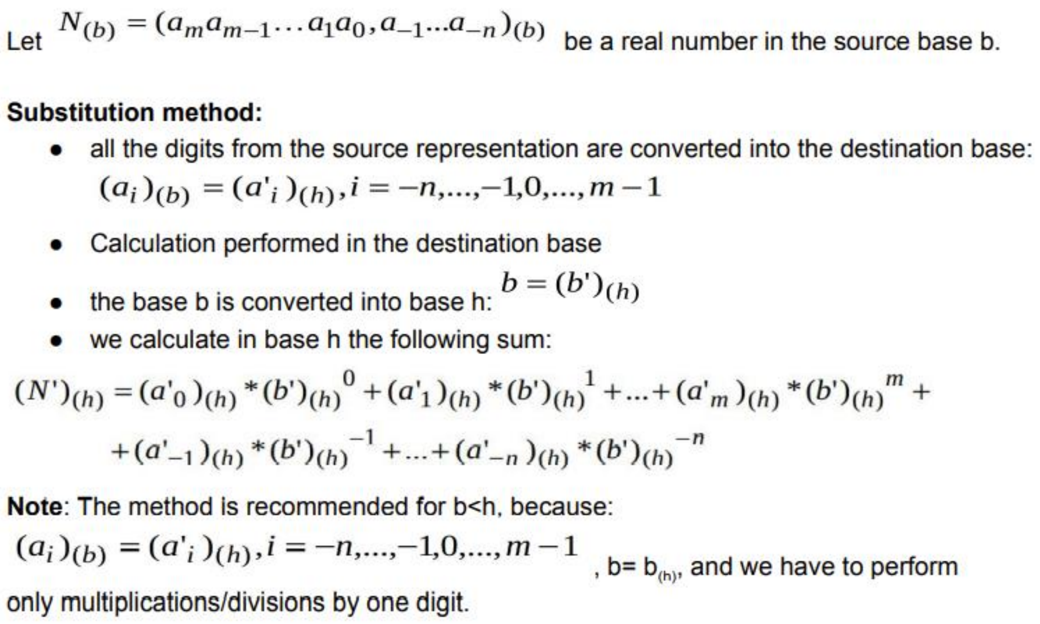
```
int mod(vector <int> number, int divisor, int base)
{
    /// This function calculates the modulo of the operation number % divisor in the specified base.
    int index, rest, number_size;
    number_size = number.size();
    rest = 0;
    for (index = 0; index < number_size; index++)
        rest = (rest * base + number[index]) % divisor;
    return rest;
}
```

The function returns the result of the operation number % divisor.

Substitution method



```
vector <int> substitution(vector <int> number, int b, int h)
{
    /// This function converts a number from base b to base h using the substitution method. (b < h)
    int power = 1, index, sum = 0;
    vector <int> result;
    std::reverse(number.begin(), number.end());
    for (index = 0; index < (int)number.size(); index++)
    {
        sum = sum + number[index] * power;
        power *= b;
    }
    while(sum)
    {
        result.push_back(sum % h);
        sum/=h;
    }
    std::reverse(result.begin(), result.end());
    return result;
}
```



Let $N_{(b)} = (a_m a_{m-1} \dots a_1 a_0 , a_{-1} \dots a_{-n})_{(b)}$ be a real number in the source base b.

Substitution method:

- all the digits from the source representation are converted into the destination base: $(a_i)_{(b)} = (a'_i)_{(h)}, i = -n, \dots, -1, 0, \dots, m-1$
- Calculation performed in the destination base
- the base b is converted into base h: $b = (b')_{(h)}$
- we calculate in base h the following sum:

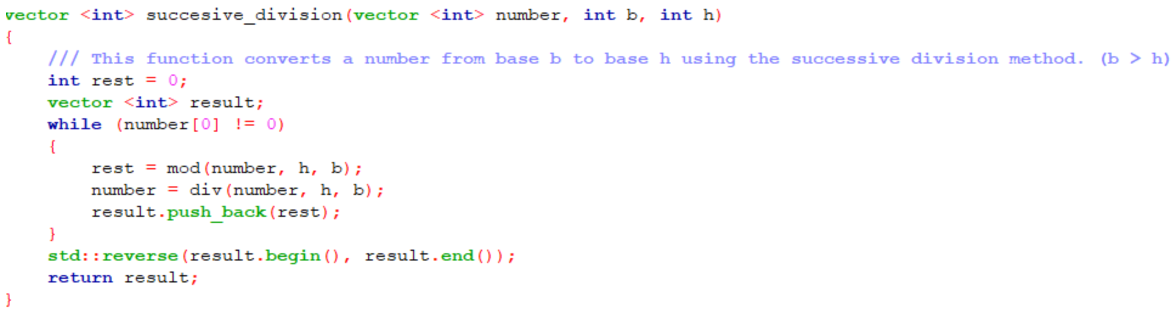
$$(N')_{(h)} = (a'_0)_{(h)} * (b')_{(h)}{}^0 + (a'_1)_{(h)} * (b')_{(h)}{}^1 + \dots + (a'_m)_{(h)} * (b')_{(h)}{}^m +$$

$$+ (a'_{-1})_{(h)} * (b')_{(h)}{}^{-1} + \dots + (a'_{-n})_{(h)} * (b')_{(h)}{}^{-n}$$

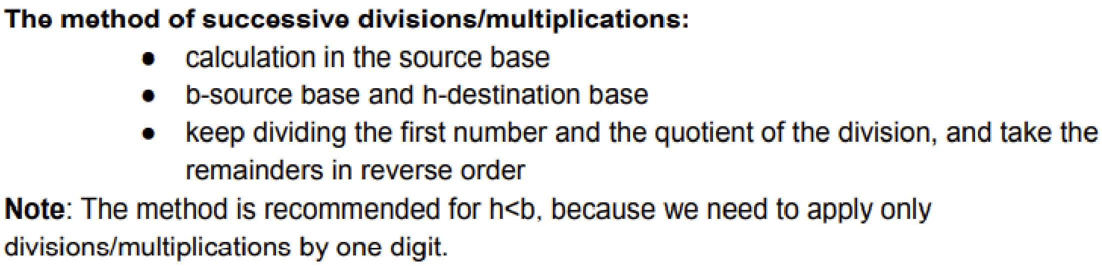
Note: The method is recommended for b<h, because: $(a_i)_{(b)} = (a'_i)_{(h)}, i = -n, \dots, -1, 0, \dots, m-1$, b= $b_{(h)}$, and we have to perform only multiplications/divisions by one digit.

Author: Lupu Eduard

Successive division method



```
vector <int> succesive_division(vector <int> number, int b, int h)
{
    /// This function converts a number from base b to base h using the successive division method. (b > h)
    int rest = 0;
    vector <int> result;
    while (number[0] != 0)
    {
        rest = mod(number, h, b);
        number = div(number, h, b);
        result.push_back(rest);
    }
    std::reverse(result.begin(), result.end());
    return result;
}
```

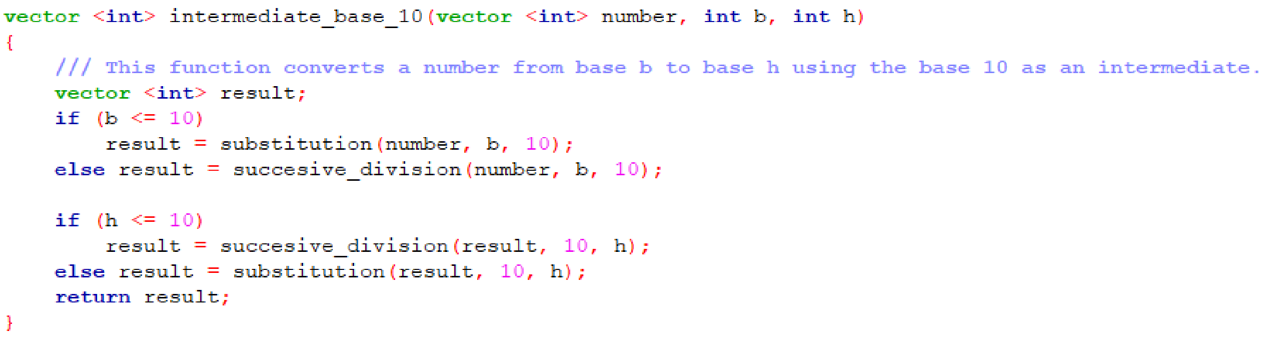


The method of successive divisions/multiplications:

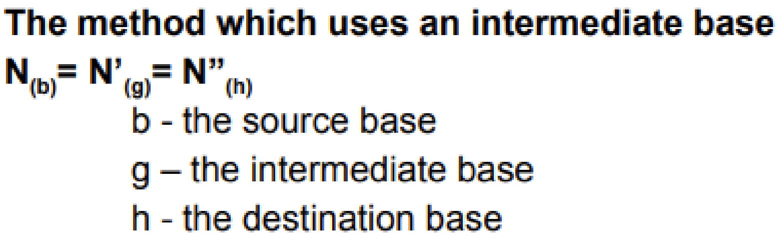
- calculation in the source base
- b-source base and h-destination base
- keep dividing the first number and the quotient of the division, and take the remainders in reverse order

Note: The method is recommended for h<b, because we need to apply only divisions/multiplications by one digit.

Intermediate base 10 method



```
vector <int> intermediate_base_10(vector <int> number, int b, int h)
{
    /// This function converts a number from base b to base h using the base 10 as an intermediate.
    vector <int> result;
    if (b <= 10)
        result = substitution(number, b, 10);
    else result = succesive_division(number, b, 10);

    if (h <= 10)
        result = succesive_division(result, 10, h);
    else result = substitution(result, 10, h);
    return result;
}
```



The method which uses an intermediate base

$N_{(b)} = N'_{(g)} = N''_{(h)}$

b - the source base

g – the intermediate base

h - the destination base