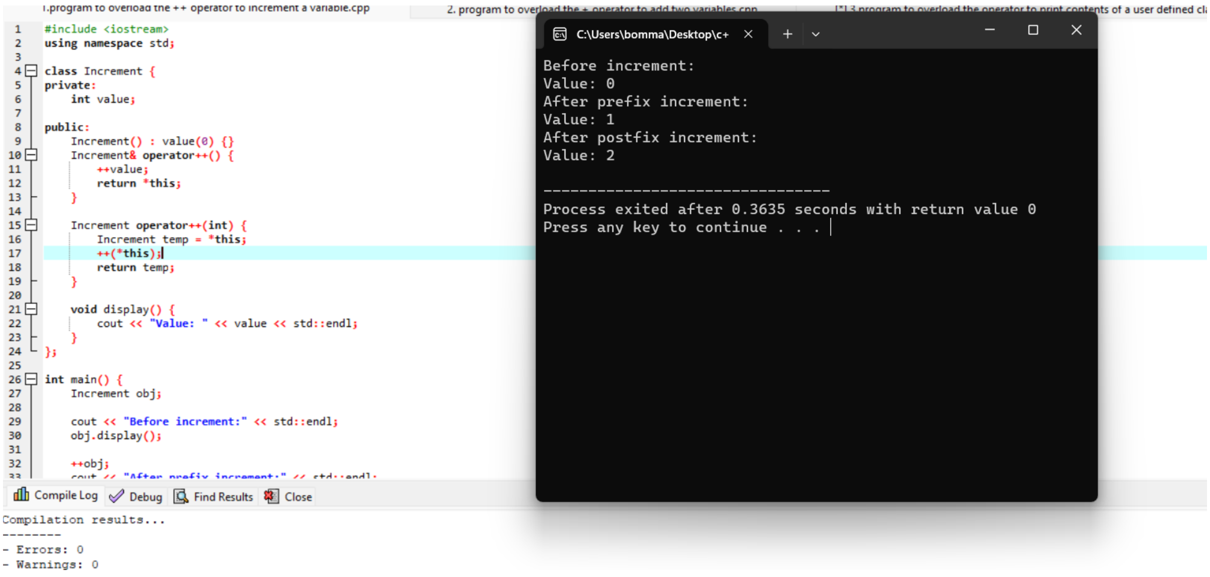
1. Write a c++ program to overload the ++ operator to increment a variable

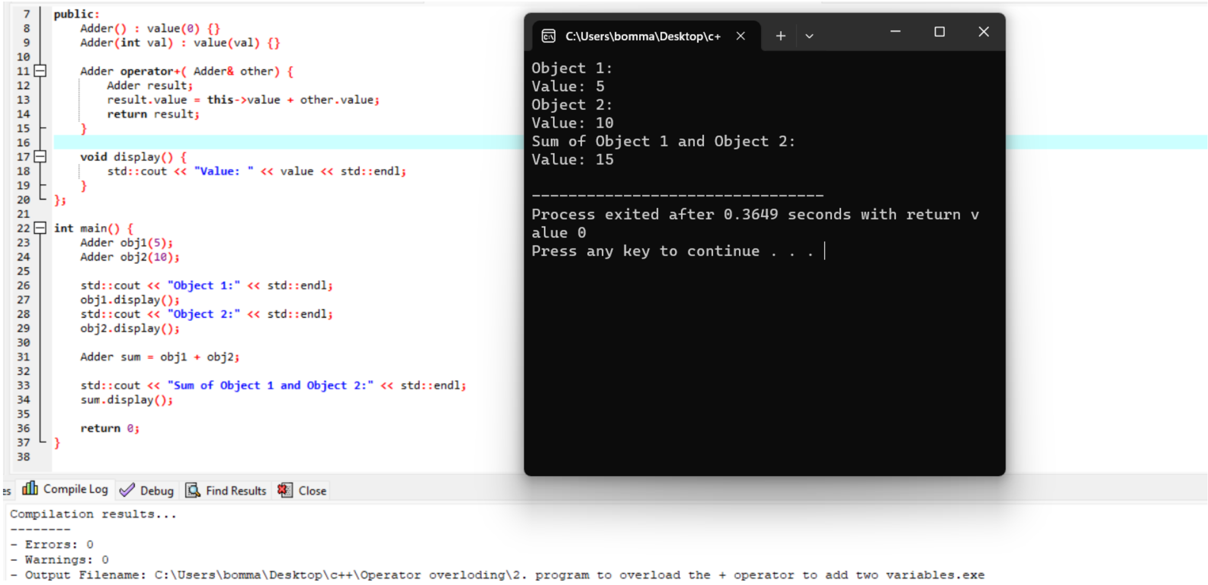

1.program to overload the ++ operator to increment a variable.cpp
2. program to overload the + operator to add two variables.cpp
[*] 3 program to overload the operator to print contents of a user defined cl
C:\Users\bomma\Desktop\c+
Compile Log
Debug
Find Results
Close

```cpp
#include <iostream>
using namespace std;

class Increment {
private:
    int value;

public:
    Increment() : value(0) {}
    Increment& operator++() {
        ++value;
        return *this;
    }

    Increment operator++(int) {
        Increment temp = *this;
        ++(*this);
        return temp;
    }

    void display() {
        cout << "Value: " << value << std::endl;
    }
};

int main() {
    Increment obj;

    cout << "Before increment:" << std::endl;
    obj.display();

    ++obj;
    cout << "After prefix increment:" << std::endl;
```

```
Before increment:
Value: 0
After prefix increment:
Value: 1
After postfix increment:
Value: 2

--------------------------------
Process exited after 0.3635 seconds with return value 0
Press any key to continue . . .
```

```
Compilation results...
--------
- Errors: 0
- Warnings: 0
```

2. Write a c++ program to overload the + operator to add two variables

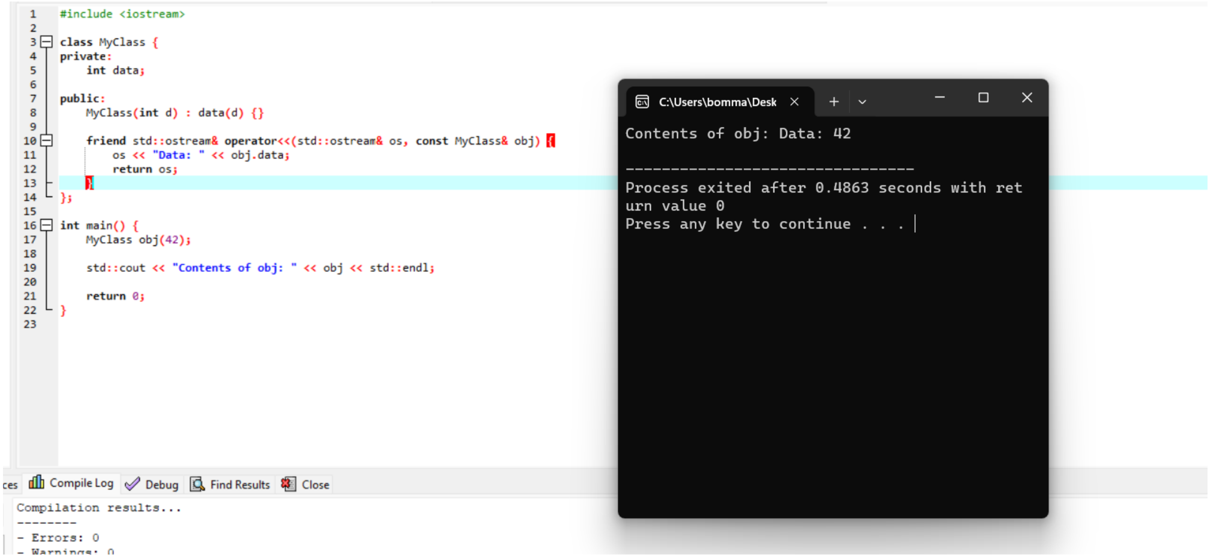


```cpp
public:
    Adder() : value(0) {}
    Adder(int val) : value(val) {}

    Adder operator+( Adder& other) {
        Adder result;
        result.value = this->value + other.value;
        return result;
    }

    void display() {
        std::cout << "Value: " << value << std::endl;
    }
};

int main() {
    Adder obj1(5);
    Adder obj2(10);

    std::cout << "Object 1:" << std::endl;
    obj1.display();
    std::cout << "Object 2:" << std::endl;
    obj2.display();

    Adder sum = obj1 + obj2;

    std::cout << "Sum of Object 1 and Object 2:" << std::endl;
    sum.display();

    return 0;
}
```

```
Object 1:
Value: 5
Object 2:
Value: 10
Sum of Object 1 and Object 2:
Value: 15

--------------------------------
Process exited after 0.3649 seconds with return v
alue 0
Press any key to continue . . .
```

```
Compilation results...
--------
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\bomma\Desktop\c++\Operator overloding\2. program to overload the + operator to add two variables.exe
```

3. Write a c++ program to overload the << operator to print contents of a user defined class

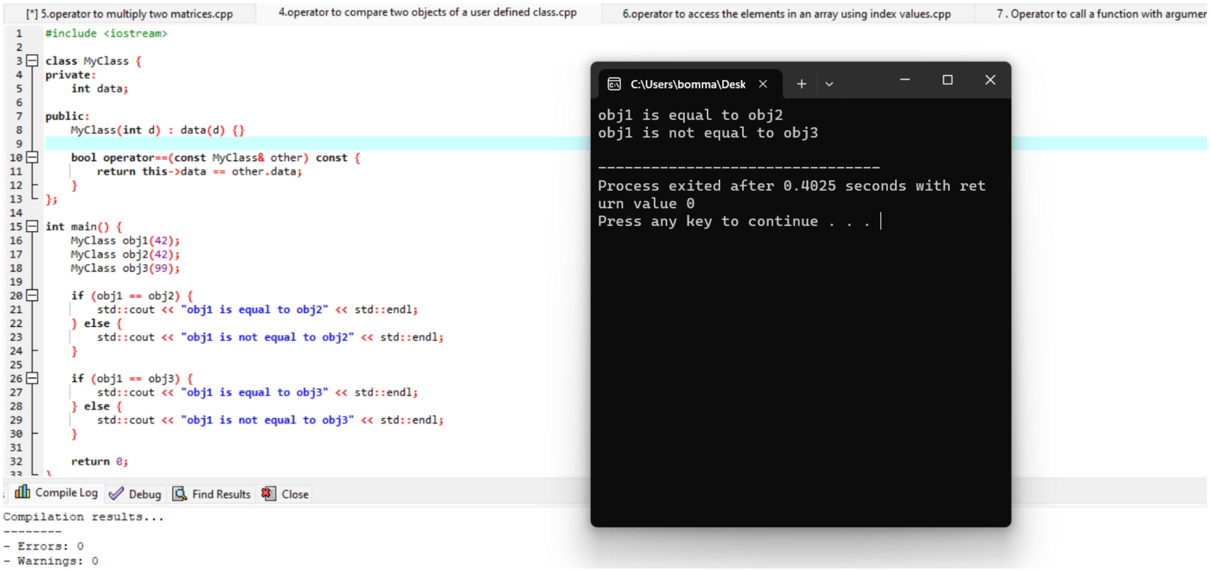


```cpp
#include <iostream>

class MyClass {
private:
    int data;

public:
    MyClass(int d) : data(d) {}

    friend std::ostream& operator<<(std::ostream& os, const MyClass& obj) {
        os << "Data: " << obj.data;
        return os;
    }
};

int main() {
    MyClass obj(42);

    std::cout << "Contents of obj: " << obj << std::endl;

    return 0;
}
```

```
Contents of obj: Data: 42

--------------------------------
Process exited after 0.4863 seconds with ret
urn value 0
Press any key to continue . . .
```

```
Compilation results...
--------
- Errors: 0
- Warnings: 0
```

4. Write a c++ program to overload the == operator to compare two objects of a user defined class

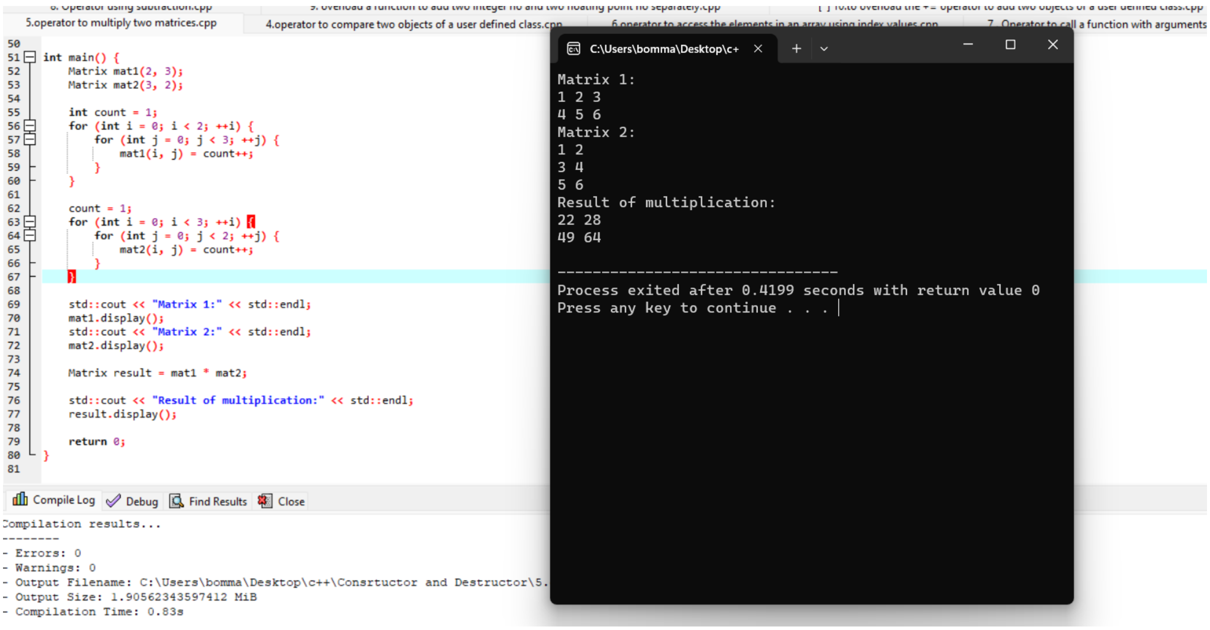

[*] 5.operator to multiply two matrices.cpp
4.operator to compare two objects of a user defined class.cpp
6.operator to access the elements in an array using index values.cpp
7 . Operator to call a function with argumer
C:\Users\bomma\Desk
Compile Log
Debug
Find Results
Close

```cpp
#include <iostream>

class MyClass {
private:
    int data;

public:
    MyClass(int d) : data(d) {}

    bool operator==(const MyClass& other) const {
        return this->data == other.data;
    }
};

int main() {
    MyClass obj1(42);
    MyClass obj2(42);
    MyClass obj3(99);

    if (obj1 == obj2) {
        std::cout << "obj1 is equal to obj2" << std::endl;
    } else {
        std::cout << "obj1 is not equal to obj2" << std::endl;
    }

    if (obj1 == obj3) {
        std::cout << "obj1 is equal to obj3" << std::endl;
    } else {
        std::cout << "obj1 is not equal to obj3" << std::endl;
    }

    return 0;
}
```

```
obj1 is equal to obj2
obj1 is not equal to obj3

--------------------------------
Process exited after 0.4025 seconds with ret
urn value 0
Press any key to continue . . .
```

```
Compilation results...
--------
- Errors: 0
- Warnings: 0
```

5. Write a c++ program to overload the * operator to multiply two matrices

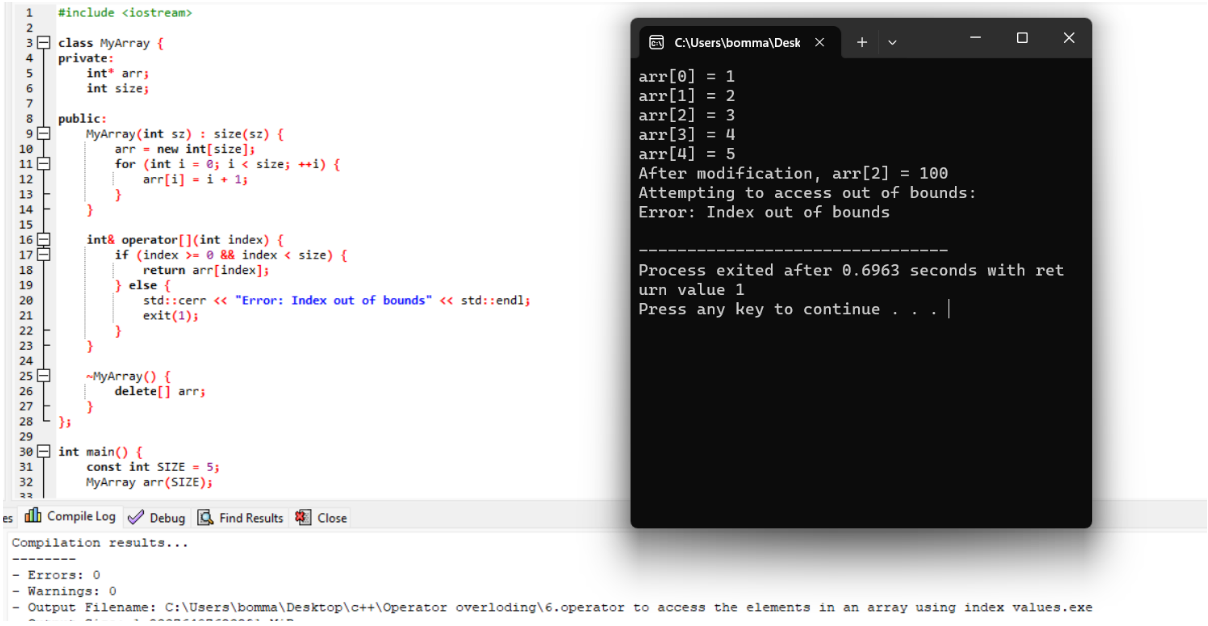

5.operator to multiply two matrices.cpp
4.operator to compare two objects of a user defined class.cpp
6.operator to access the elements in an array using index values.cpp
7. Operator to call a function with arguments
C:\Users\bomma\Desktop\c+
Compile Log
Debug
Find Results
Close

```cpp
int main() {
    Matrix mat1(2, 3);
    Matrix mat2(3, 2);

    int count = 1;
    for (int i = 0; i < 2; ++i) {
        for (int j = 0; j < 3; ++j) {
            mat1(i, j) = count++;
        }
    }

    count = 1;
    for (int i = 0; i < 3; ++i) {
        for (int j = 0; j < 2; ++j) {
            mat2(i, j) = count++;
        }
    }

    std::cout << "Matrix 1:" << std::endl;
    mat1.display();
    std::cout << "Matrix 2:" << std::endl;
    mat2.display();

    Matrix result = mat1 * mat2;

    std::cout << "Result of multiplication:" << std::endl;
    result.display();

    return 0;
}
```

```
Matrix 1:
1 2 3
4 5 6
Matrix 2:
1 2
3 4
5 6
Result of multiplication:
22 28
49 64

--------------------------------
Process exited after 0.4199 seconds with return value 0
Press any key to continue . . .
```

```
Compilation results...
--------
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\bomma\Desktop\c++\Consrtuctor and Destructor\5.
- Output Size: 1.90562343597412 MiB
- Compilation Time: 0.83s
```

6. write a c++ program to overload the [] operator to access the elements in an array using index values

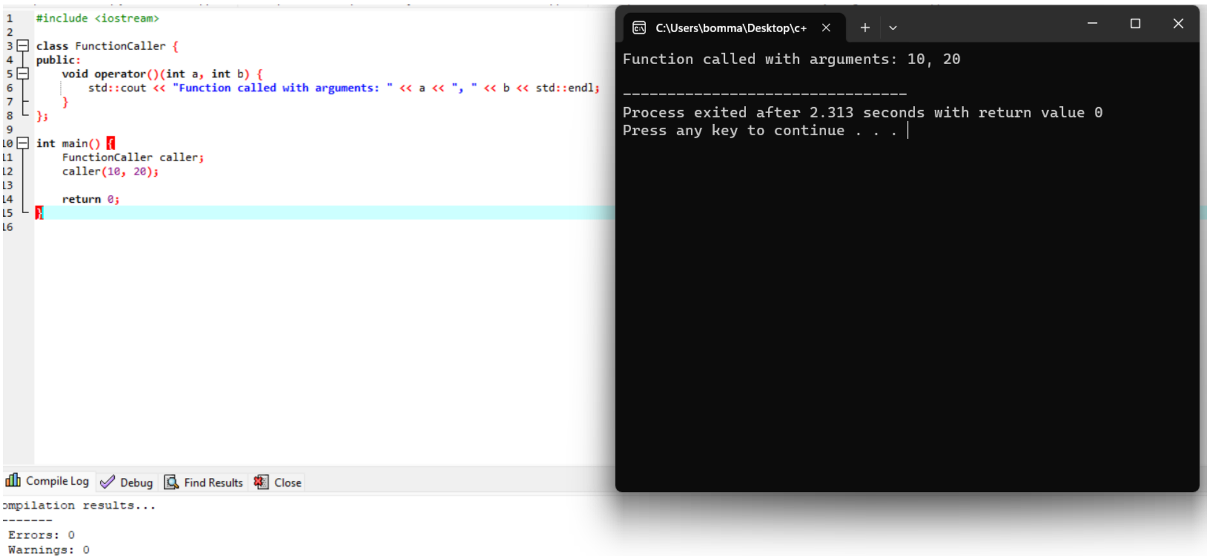


```cpp
#include <iostream>

class MyArray {
private:
    int* arr;
    int size;

public:
    MyArray(int sz) : size(sz) {
        arr = new int[size];
        for (int i = 0; i < size; ++i) {
            arr[i] = i + 1;
        }
    }

    int& operator[](int index) {
        if (index >= 0 && index < size) {
            return arr[index];
        } else {
            std::cerr << "Error: Index out of bounds" << std::endl;
            exit(1);
        }
    }

    ~MyArray() {
        delete[] arr;
    }
};

int main() {
    const int SIZE = 5;
    MyArray arr(SIZE);
```

```
arr[0] = 1
arr[1] = 2
arr[2] = 3
arr[3] = 4
arr[4] = 5
After modification, arr[2] = 100
Attempting to access out of bounds:
Error: Index out of bounds

--------------------------------
Process exited after 0.6963 seconds with ret
urn value 1
Press any key to continue . . .
```

```
Compilation results...
--------
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\bomma\Desktop\c++\Operator overloding\6.operator to access the elements in an array using index values.exe
```

7. Write a c++ program to overload the () to call a function with arguments

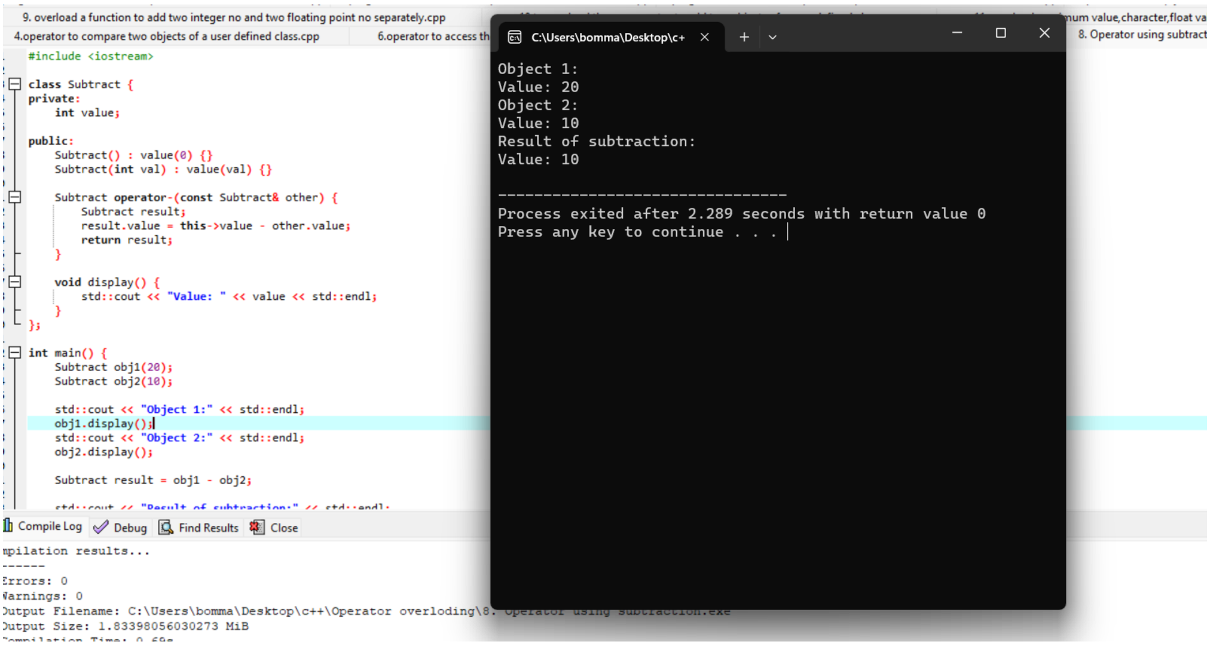


```
#include <iostream>

class FunctionCaller {
public:
    void operator()(int a, int b) {
        std::cout << "Function called with arguments: " << a << ", " << b << std::endl;
    }
};

int main() {
    FunctionCaller caller;
    caller(10, 20);

    return 0;
}
```

```
Function called with arguments: 10, 20

--------------------------------
Process exited after 2.313 seconds with return value 0
Press any key to continue . . .
```

```
ompilation results...
--------
 Errors: 0
 Warnings: 0
```

8.write a c++ program to overload the – operator to subtract two variables



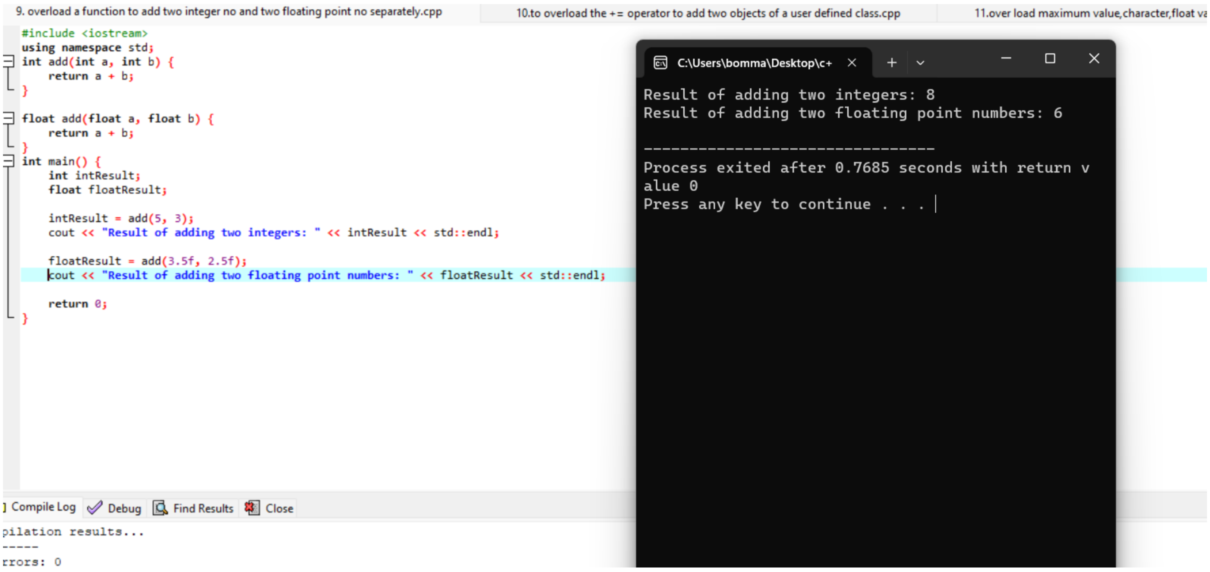
9. overload a function to add two integer no and two floating point no separately.cpp
4.operator to compare two objects of a user defined class.cpp
6.operator to access th
mum value,character,float va
8. Operator using subtract
C:\Users\bomma\Desktop\c+
Compile Log
Debug
Find Results
Close

```
#include <iostream>

class Subtract {
private:
    int value;

public:
    Subtract() : value(0) {}
    Subtract(int val) : value(val) {}

    Subtract operator-(const Subtract& other) {
        Subtract result;
        result.value = this->value - other.value;
        return result;
    }

    void display() {
        std::cout << "Value: " << value << std::endl;
    }
};

int main() {
    Subtract obj1(20);
    Subtract obj2(10);

    std::cout << "Object 1:" << std::endl;
    obj1.display();
    std::cout << "Object 2:" << std::endl;
    obj2.display();

    Subtract result = obj1 - obj2;

    std::cout << "Result of subtraction:" << std::endl;
```

```
Object 1:
Value: 20
Object 2:
Value: 10
Result of subtraction:
Value: 10

--------------------------------
Process exited after 2.289 seconds with return value 0
Press any key to continue . . .
```

```
mpilation results...
------
Errors: 0
Warnings: 0
Output Filename: C:\Users\bomma\Desktop\c++\Operator overloding\8. Operator using subtraction.exe
Output Size: 1.83398056030273 MiB
Compilation Time: 0.69s
```

9.write a c++ program to overload a function to add two integer numbers and two floating point number separately



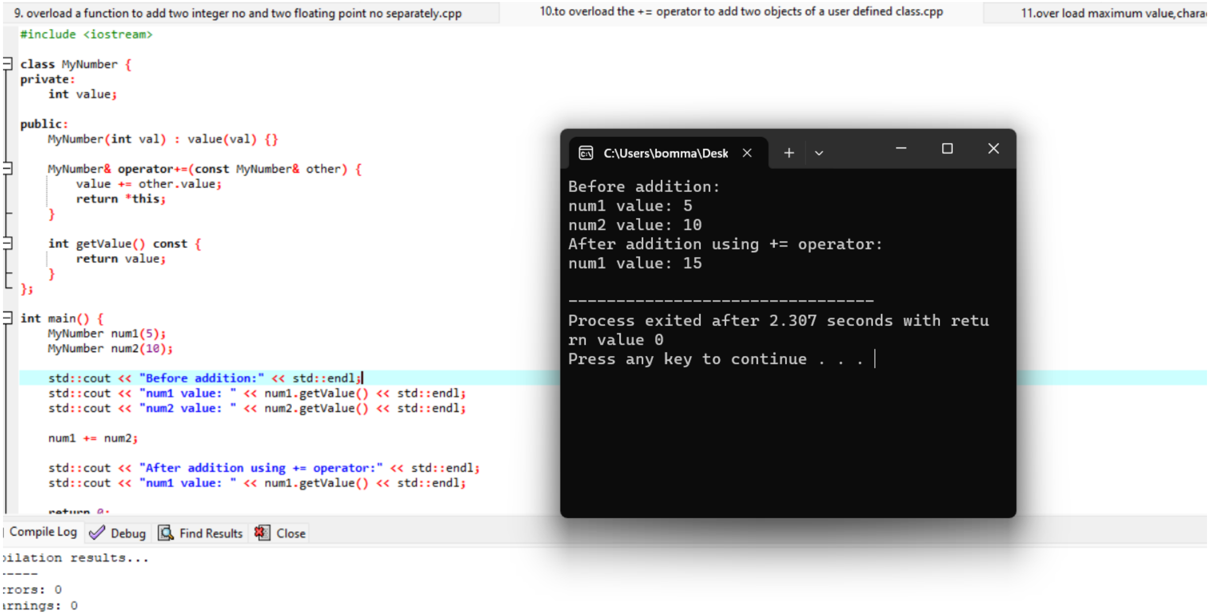
9. overload a function to add two integer no and two floating point no separately.cpp
10.to overload the += operator to add two objects of a user defined class.cpp
11.over load maximum value,character,float va
C:\Users\bomma\Desktop\c+
Compile Log
Debug
Find Results
Close

```cpp
#include <iostream>
using namespace std;
int add(int a, int b) {
    return a + b;
}

float add(float a, float b) {
    return a + b;
}
int main() {
    int intResult;
    float floatResult;

    intResult = add(5, 3);
    cout << "Result of adding two integers: " << intResult << std::endl;

    floatResult = add(3.5f, 2.5f);
    cout << "Result of adding two floating point numbers: " << floatResult << std::endl;

    return 0;
}
```

```
Result of adding two integers: 8
Result of adding two floating point numbers: 6

--------------------------------
Process exited after 0.7685 seconds with return v
alue 0
Press any key to continue . . .
```

```
pilation results...
-----
rrors: 0
```

10.Write a c++ program to overload the += operator to add two objects of a user defined class



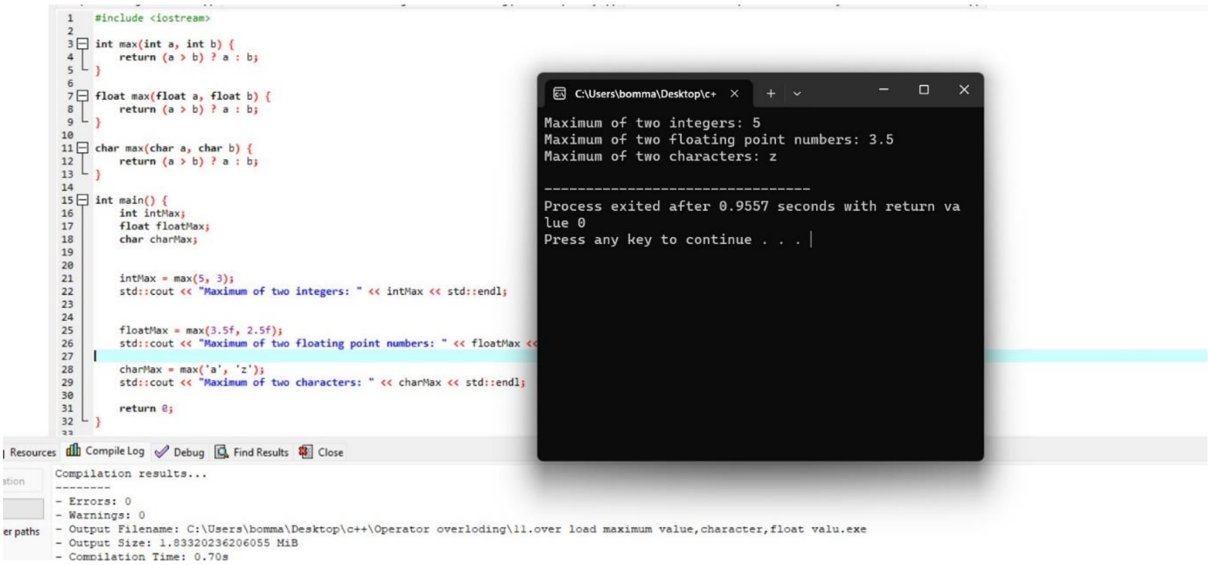
9. overload a function to add two integer no and two floating point no separately.cpp
10.to overload the += operator to add two objects of a user defined class.cpp
11.over load maximum value,chara
C:\Users\bomma\Desk
Compile Log
Debug
Find Results
Close

```cpp
#include <iostream>

class MyNumber {
private:
    int value;

public:
    MyNumber(int val) : value(val) {}

    MyNumber& operator+=(const MyNumber& other) {
        value += other.value;
        return *this;
    }

    int getValue() const {
        return value;
    }
};

int main() {
    MyNumber num1(5);
    MyNumber num2(10);

    std::cout << "Before addition:" << std::endl;
    std::cout << "num1 value: " << num1.getValue() << std::endl;
    std::cout << "num2 value: " << num2.getValue() << std::endl;

    num1 += num2;

    std::cout << "After addition using += operator:" << std::endl;
    std::cout << "num1 value: " << num1.getValue() << std::endl;

    return 0;
```

```
Before addition:
num1 value: 5
num2 value: 10
After addition using += operator:
num1 value: 15

--------------------------------
Process exited after 2.307 seconds with retu
rn value 0
Press any key to continue . . .
```

```
pilation results...
-----
rors: 0
rnings: 0
```

11.write a c++ program to overload a function to find the maximum value from two integer numbers and two floating point number, and two characters separately

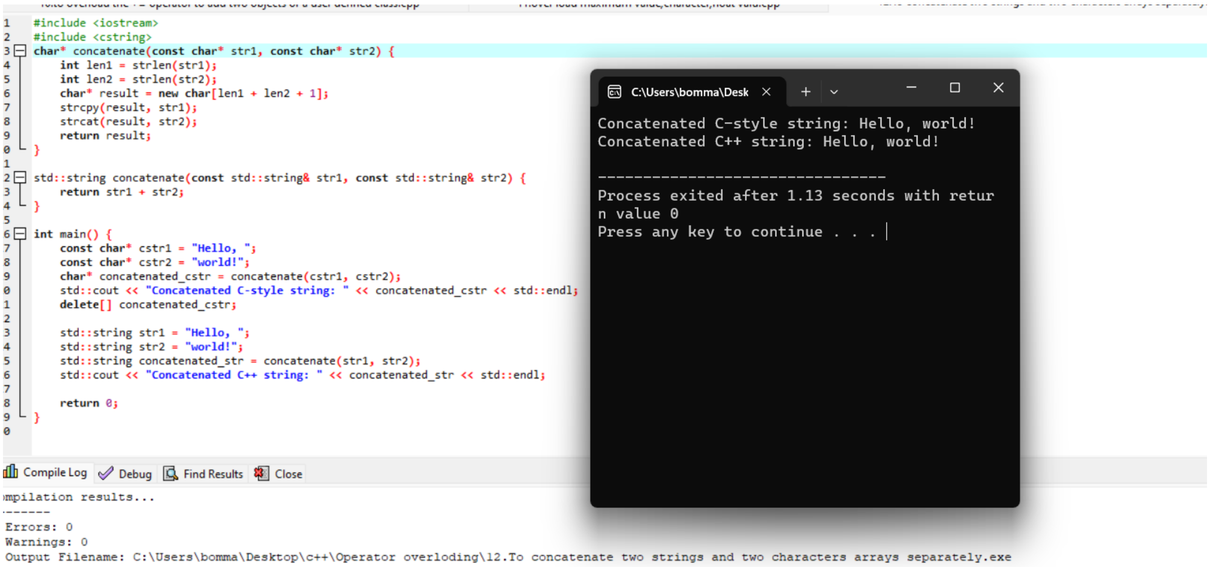


```cpp
#include <iostream>

int max(int a, int b) {
    return (a > b) ? a : b;
}

float max(float a, float b) {
    return (a > b) ? a : b;
}

char max(char a, char b) {
    return (a > b) ? a : b;
}

int main() {
    int intMax;
    float floatMax;
    char charMax;


    intMax = max(5, 3);
    std::cout << "Maximum of two integers: " << intMax << std::endl;


    floatMax = max(3.5f, 2.5f);
    std::cout << "Maximum of two floating point numbers: " << floatMax <<

    charMax = max('a', 'z');
    std::cout << "Maximum of two characters: " << charMax << std::endl;

    return 0;
}
```

```
Maximum of two integers: 5
Maximum of two floating point numbers: 3.5
Maximum of two characters: z

--------------------------------
Process exited after 0.9557 seconds with return va
lue 0
Press any key to continue . . .
```

```
Compilation results...
--------
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\bomma\Desktop\c++\Operator overloding\11.over load maximum value,character,float valu.exe
- Output Size: 1.83320236206055 MiB
- Compilation Time: 0.70s
```

12.write a c++ program to overload a function to concatenate two strings and two characters arrays separately

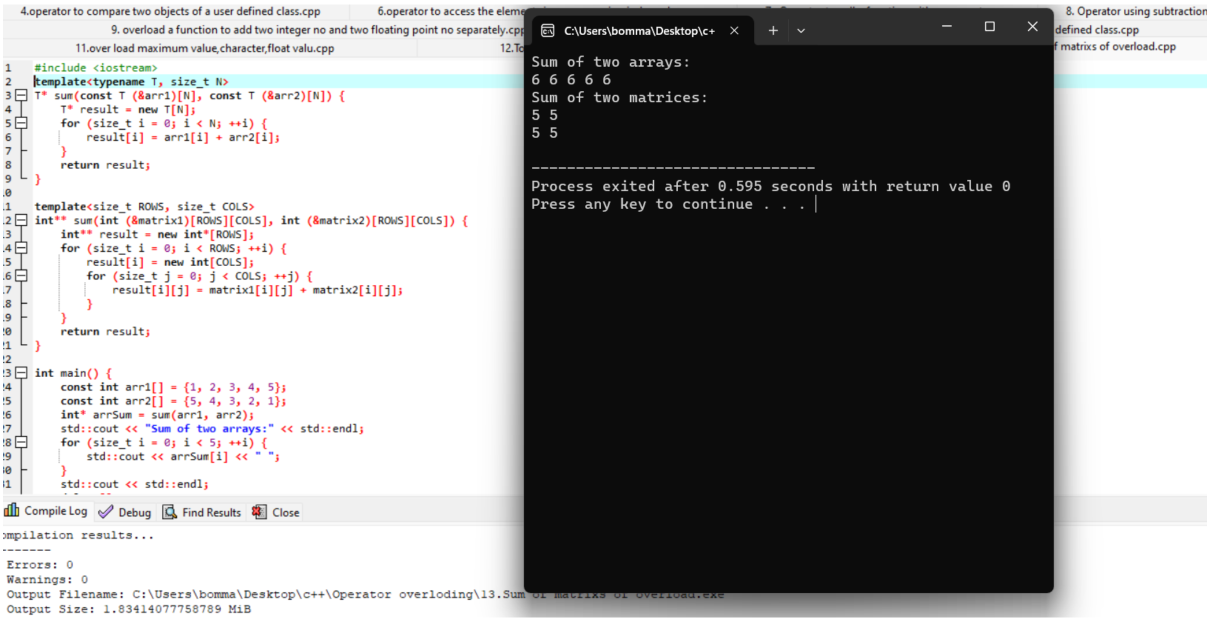


```cpp
#include <iostream>
#include <cstring>
char* concatenate(const char* str1, const char* str2) {
    int len1 = strlen(str1);
    int len2 = strlen(str2);
    char* result = new char[len1 + len2 + 1];
    strcpy(result, str1);
    strcat(result, str2);
    return result;
}

std::string concatenate(const std::string& str1, const std::string& str2) {
    return str1 + str2;
}

int main() {
    const char* cstr1 = "Hello, ";
    const char* cstr2 = "world!";
    char* concatenated_cstr = concatenate(cstr1, cstr2);
    std::cout << "Concatenated C-style string: " << concatenated_cstr << std::endl;
    delete[] concatenated_cstr;

    std::string str1 = "Hello, ";
    std::string str2 = "world!";
    std::string concatenated_str = concatenate(str1, str2);
    std::cout << "Concatenated C++ string: " << concatenated_str << std::endl;

    return 0;
}
```

```
Concatenated C-style string: Hello, world!
Concatenated C++ string: Hello, world!

--------------------------------
Process exited after 1.13 seconds with retur
n value 0
Press any key to continue . . .
```

```
mpilation results...
-------
Errors: 0
Warnings: 0
Output Filename: C:\Users\bomma\Desktop\c++\Operator overloding\12.To concatenate two strings and two characters arrays separately.exe
```

13.write a c++ program to overload a function to calculate the sum of two matrices and two arrays separately

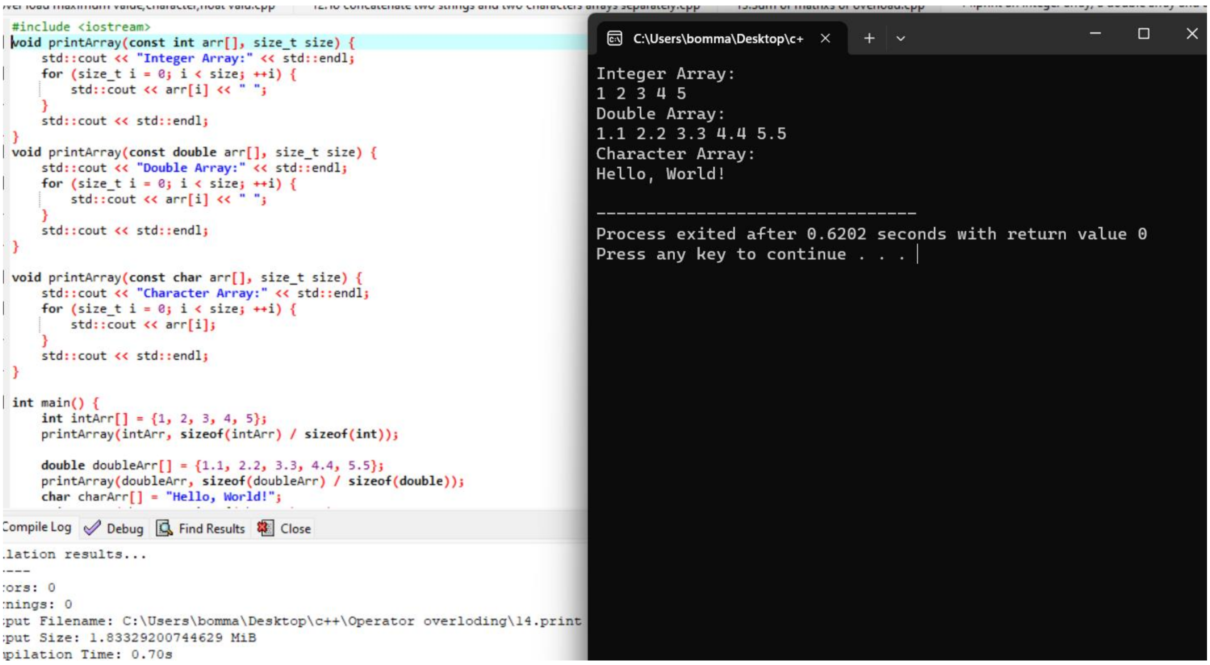

4.operator to compare two objects of a user defined class.cpp
6.operator to access the eleme
9. overload a function to add two integer no and two floating point no separately.cpp
11.over load maximum value,character,float valu.cpp
12.To
8. Operator using subtraction
defined class.cpp
f matrixs of overload.cpp
C:\Users\bomma\Desktop\c+
Compile Log
Debug
Find Results
Close

```cpp
#include <iostream>
template<typename T, size_t N>
T* sum(const T (&arr1)[N], const T (&arr2)[N]) {
    T* result = new T[N];
    for (size_t i = 0; i < N; ++i) {
        result[i] = arr1[i] + arr2[i];
    }
    return result;
}

template<size_t ROWS, size_t COLS>
int** sum(int (&matrix1)[ROWS][COLS], int (&matrix2)[ROWS][COLS]) {
    int** result = new int*[ROWS];
    for (size_t i = 0; i < ROWS; ++i) {
        result[i] = new int[COLS];
        for (size_t j = 0; j < COLS; ++j) {
            result[i][j] = matrix1[i][j] + matrix2[i][j];
        }
    }
    return result;
}

int main() {
    const int arr1[] = {1, 2, 3, 4, 5};
    const int arr2[] = {5, 4, 3, 2, 1};
    int* arrSum = sum(arr1, arr2);
    std::cout << "Sum of two arrays:" << std::endl;
    for (size_t i = 0; i < 5; ++i) {
        std::cout << arrSum[i] << " ";
    }
    std::cout << std::endl;
```

```
ompilation results...
-------
 Errors: 0
 Warnings: 0
 Output Filename: C:\Users\bomma\Desktop\c++\Operator overloding\13.Sum of matrixs of overload.exe
 Output Size: 1.83414077758789 MiB
```

```
Sum of two arrays:
6 6 6 6 6
Sum of two matrices:
5 5
5 5

--------------------------------
Process exited after 0.595 seconds with return value 0
Press any key to continue . . .
```

14. write a c++ program to overload a function to print an integer array, a double array and a character array separately

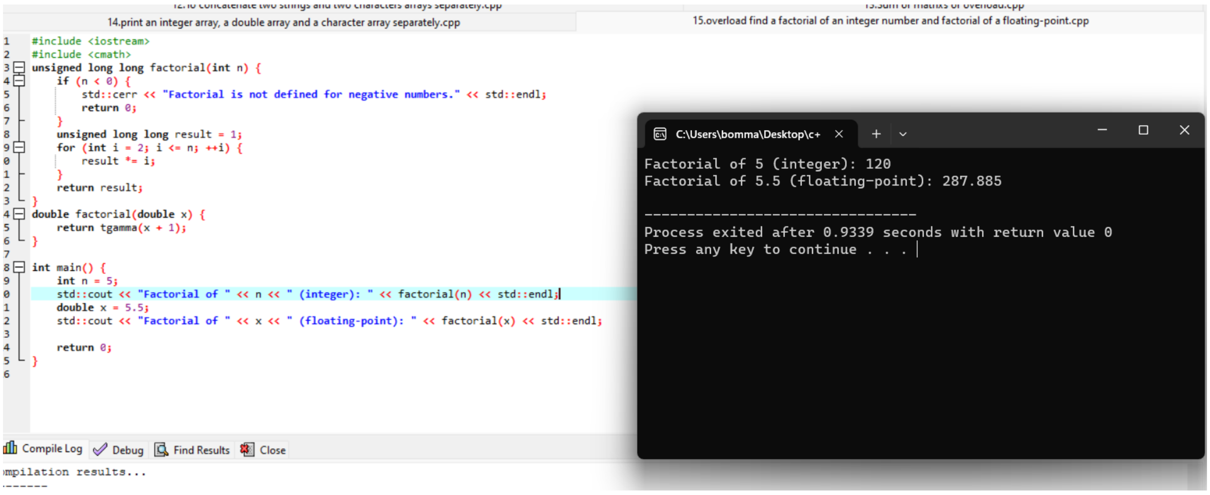

ver load maximum value,character,float valu.cpp
12.To concatenate two strings and two characters arrays separately.cpp
13.Sum of matrixs of overload.cpp
C:\Users\bomma\Desktop\c+
Compile Log
Debug
Find Results
Close

```cpp
#include <iostream>
void printArray(const int arr[], size_t size) {
    std::cout << "Integer Array:" << std::endl;
    for (size_t i = 0; i < size; ++i) {
        std::cout << arr[i] << " ";
    }
    std::cout << std::endl;
}
void printArray(const double arr[], size_t size) {
    std::cout << "Double Array:" << std::endl;
    for (size_t i = 0; i < size; ++i) {
        std::cout << arr[i] << " ";
    }
    std::cout << std::endl;
}

void printArray(const char arr[], size_t size) {
    std::cout << "Character Array:" << std::endl;
    for (size_t i = 0; i < size; ++i) {
        std::cout << arr[i];
    }
    std::cout << std::endl;
}

int main() {
    int intArr[] = {1, 2, 3, 4, 5};
    printArray(intArr, sizeof(intArr) / sizeof(int));

    double doubleArr[] = {1.1, 2.2, 3.3, 4.4, 5.5};
    printArray(doubleArr, sizeof(doubleArr) / sizeof(double));
    char charArr[] = "Hello, World!";
```

```
.lation results...
----
:ors: 0
:nings: 0
:put Filename: C:\Users\bomma\Desktop\c++\Operator overloding\14.print
:put Size: 1.83329200744629 MiB
ıpilation Time: 0.70s
```

```
Integer Array:
1 2 3 4 5
Double Array:
1.1 2.2 3.3 4.4 5.5
Character Array:
Hello, World!

--------------------------------
Process exited after 0.6202 seconds with return value 0
Press any key to continue . . .
```

15. write a c++ program to overload a function to find a factorial of an integer number and factorial of a floating-point number separately



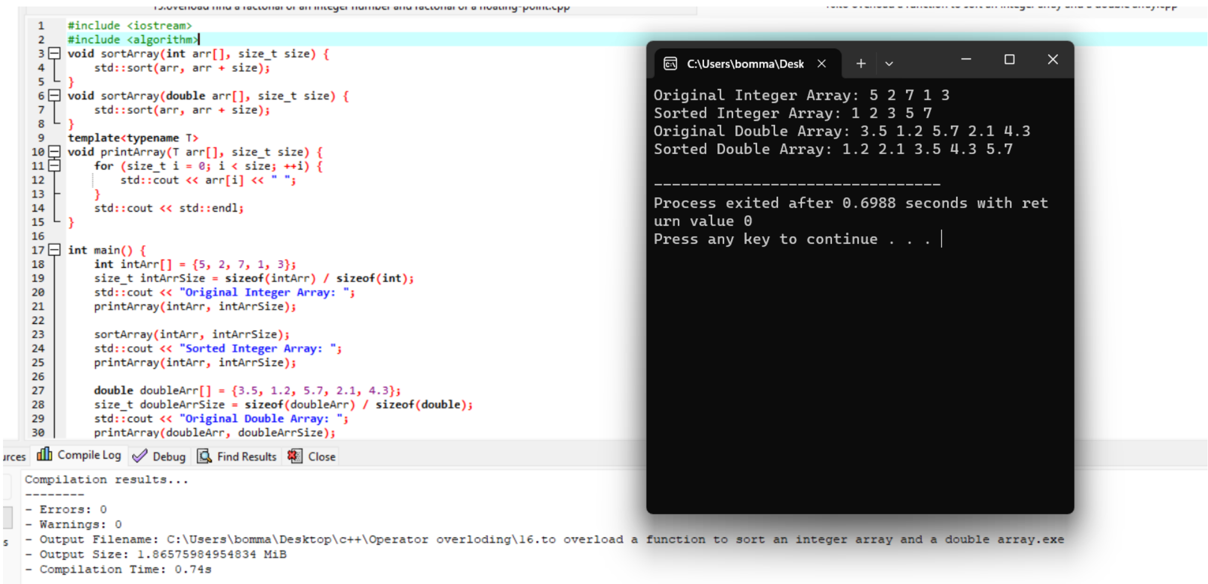
14.print an integer array, a double array and a character array separately.cpp
15.overload find a factorial of an integer number and factorial of a floating-point.cpp
Compile Log
Debug
Find Results
Close
C:\Users\bomma\Desktop\c+

```cpp
#include <iostream>
#include <cmath>
unsigned long long factorial(int n) {
    if (n < 0) {
        std::cerr << "Factorial is not defined for negative numbers." << std::endl;
        return 0;
    }
    unsigned long long result = 1;
    for (int i = 2; i <= n; ++i) {
        result *= i;
    }
    return result;
}
double factorial(double x) {
    return tgamma(x + 1);
}

int main() {
    int n = 5;
    std::cout << "Factorial of " << n << " (integer): " << factorial(n) << std::endl;
    double x = 5.5;
    std::cout << "Factorial of " << x << " (floating-point): " << factorial(x) << std::endl;

    return 0;
}
```

```
Factorial of 5 (integer): 120
Factorial of 5.5 (floating-point): 287.885

--------------------------------
Process exited after 0.9339 seconds with return value 0
Press any key to continue . . .
```

```
mpilation results...
-------
```

16. write a c++ program to overload a function to sort an integer array and a double array



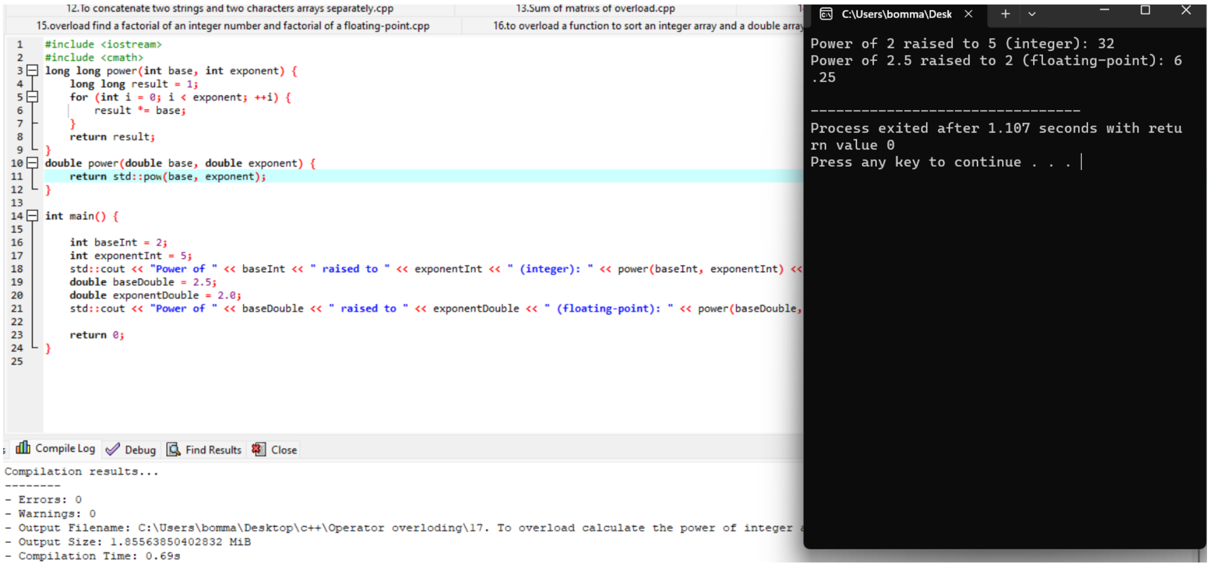
15.overload find a factorial of an integer number and factorial of a floating-point.cpp
Compile Log
Debug
Find Results
Close
C:\Users\bomma\Desk

```cpp
#include <iostream>
#include <algorithm>
void sortArray(int arr[], size_t size) {
    std::sort(arr, arr + size);
}
void sortArray(double arr[], size_t size) {
    std::sort(arr, arr + size);
}
template<typename T>
void printArray(T arr[], size_t size) {
    for (size_t i = 0; i < size; ++i) {
        std::cout << arr[i] << " ";
    }
    std::cout << std::endl;
}

int main() {
    int intArr[] = {5, 2, 7, 1, 3};
    size_t intArrSize = sizeof(intArr) / sizeof(int);
    std::cout << "Original Integer Array: ";
    printArray(intArr, intArrSize);

    sortArray(intArr, intArrSize);
    std::cout << "Sorted Integer Array: ";
    printArray(intArr, intArrSize);

    double doubleArr[] = {3.5, 1.2, 5.7, 2.1, 4.3};
    size_t doubleArrSize = sizeof(doubleArr) / sizeof(double);
    std::cout << "Original Double Array: ";
    printArray(doubleArr, doubleArrSize);
```

```
Original Integer Array: 5 2 7 1 3
Sorted Integer Array: 1 2 3 5 7
Original Double Array: 3.5 1.2 5.7 2.1 4.3
Sorted Double Array: 1.2 2.1 3.5 4.3 5.7

--------------------------------
Process exited after 0.6988 seconds with ret
urn value 0
Press any key to continue . . .
```

```
Compilation results...
--------
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\bomma\Desktop\c++\Operator overloding\16.to overload a function to sort an integer array and a double array.exe
- Output Size: 1.86575984954834 MiB
- Compilation Time: 0.74s
```

17. write a c++ program to overload a function to calculate the power of an integer number and power of a floating-point number separately

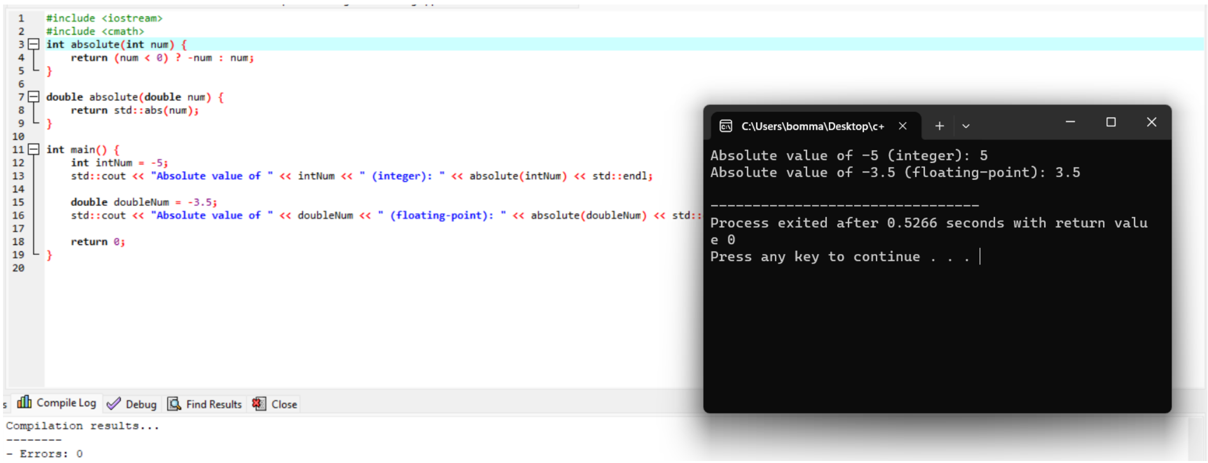

12.To concatenate two strings and two characters arrays separately.cpp
13.Sum of matrixs of overload.cpp
15.overload find a factorial of an integer number and factorial of a floating-point.cpp
16.to overload a function to sort an integer array and a double array
C:\Users\bomma\Desk
Compile Log
Debug
Find Results
Close

```cpp
#include <iostream>
#include <cmath>
long long power(int base, int exponent) {
    long long result = 1;
    for (int i = 0; i < exponent; ++i) {
        result *= base;
    }
    return result;
}
double power(double base, double exponent) {
    return std::pow(base, exponent);
}

int main() {

    int baseInt = 2;
    int exponentInt = 5;
    std::cout << "Power of " << baseInt << " raised to " << exponentInt << " (integer): " << power(baseInt, exponentInt) <<
    double baseDouble = 2.5;
    double exponentDouble = 2.0;
    std::cout << "Power of " << baseDouble << " raised to " << exponentDouble << " (floating-point): " << power(baseDouble,

    return 0;
}
```

```
Compilation results...
--------
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\bomma\Desktop\c++\Operator overloding\17. To overload calculate the power of integer a
- Output Size: 1.85563850402832 MiB
- Compilation Time: 0.69s
```

```
Power of 2 raised to 5 (integer): 32
Power of 2.5 raised to 2 (floating-point): 6
.25

--------------------------------
Process exited after 1.107 seconds with retu
rn value 0
Press any key to continue . . .
```

18. write a c++ program to overload a function to find an absolute value of an integer number and absolute value of a floating-point number separately



```cpp
#include <iostream>
#include <cmath>
int absolute(int num) {
    return (num < 0) ? -num : num;
}

double absolute(double num) {
    return std::abs(num);
}

int main() {
    int intNum = -5;
    std::cout << "Absolute value of " << intNum << " (integer): " << absolute(intNum) << std::endl;

    double doubleNum = -3.5;
    std::cout << "Absolute value of " << doubleNum << " (floating-point): " << absolute(doubleNum) << std::

    return 0;
}
```

```
Compilation results...
--------
- Errors: 0
```

```
Absolute value of -5 (integer): 5
Absolute value of -3.5 (floating-point): 3.5

--------------------------------
Process exited after 0.5266 seconds with return valu
e 0
Press any key to continue . . .
```