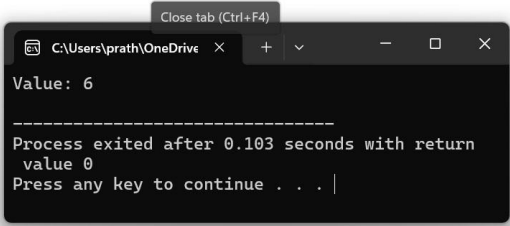


1. Write a c++ program to overload the ++ operator to increment a variable

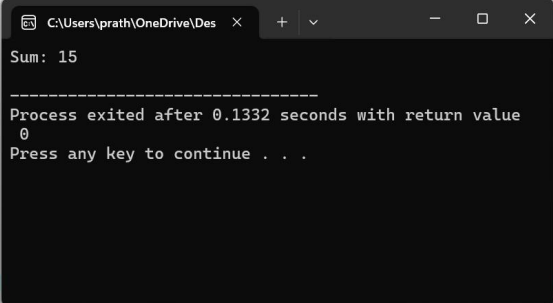
```
1. ++ operator to increment a variable.cpp
1 #include <iostream>
2 using namespace std;
3
4 class Increment {
5     int value;
6 public:
7     Increment(int v) : value(v) {}
8
9     void operator++() {
10         ++value;
11     }
12
13     void display() {
14         cout << "Value: " << value << endl;
15     }
16 };
17
18 int main() {
19     Increment obj(5);
20     ++obj;
21     obj.display();
22     return 0;
23 }
24
```



```
Close tab (Ctrl+F4)
C:\Users\prath\OneDrive
Value: 6
-----
Process exited after 0.103 seconds with return
value 0
Press any key to continue . . . |
```

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

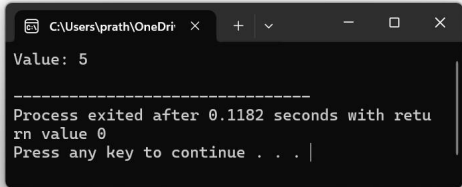
```
2. + operator to add two variables.cpp
1 #include <iostream>
2 using namespace std;
3
4 class Add {
5     int value;
6 public:
7     Add(int v) : value(v) {}
8
9     Add operator+(const Add &obj) {
10         return Add(value + obj.value);
11     }
12
13     void display() {
14         cout << "Sum: " << value << endl;
15     }
16 };
17
18 int main() {
19     Add obj1(5), obj2(10);
20     Add obj3 = obj1 + obj2;
21     obj3.display();
22     return 0;
23 }
24
```



```
C:\Users\prath\OneDrive\Des
Sum: 15
-----
Process exited after 0.1332 seconds with return value
0
Press any key to continue . . .
```

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

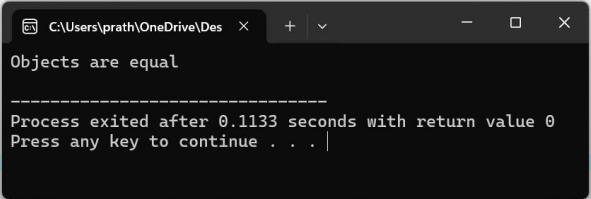
```
3. les than les than operator to print contents of a user defined class.cpp
1 #include <iostream>
2 using namespace std;
3
4 class Print {
5     int value;
6 public:
7     Print(int v) : value(v) {}
8
9     friend ostream& operator<<(ostream &out, const Print &obj);
10 };
11
12 ostream& operator<<(ostream &out, const Print &obj) {
13     out << "Value: " << obj.value;
14     return out;
15 }
16
17 int main() {
18     Print obj(5);
19     cout << obj << endl;
20     return 0;
21 }
22
```



```
C:\Users\prath\OneDri
Value: 5
-----
Process exited after 0.1182 seconds with retu
rn value 0
Press any key to continue . . . |
```

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

```
4. == operator to compare two objects of a user defined class.cpp
1 #include <iostream>
2 using namespace std;
3
4 class Compare {
5     int value;
6 public:
7     Compare(int v) : value(v) {}
8
9     bool operator==(const Compare &obj) {
10         return value == obj.value;
11     }
12 };
13
14 int main() {
15     Compare obj1(5), obj2(5);
16     if (obj1 == obj2)
17         cout << "Objects are equal" << endl;
18     else
19         cout << "Objects are not equal" << endl;
20     return 0;
21 }
22
```



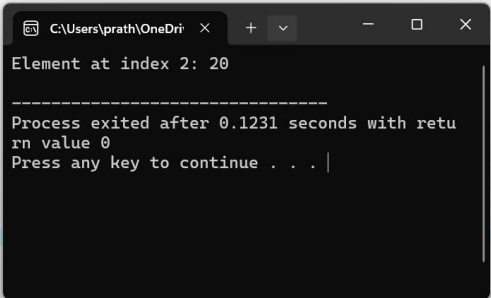
5. Write a c++ program to overload the * operator to multiply two matrices

```
5. operator to multiply two matrices.cpp
1 #include <iostream>
2 using namespace std;
3
4 class Matrix {
5     int mat[2][2];
6 public:
7     Matrix(int a, int b, int c, int d) {
8         mat[0][0] = a; mat[0][1] = b;
9         mat[1][0] = c; mat[1][1] = d;
10    }
11
12    Matrix operator*(const Matrix &m) {
13        Matrix temp(0, 0, 0, 0);
14        for (int i = 0; i < 2; ++i)
15            for (int j = 0; j < 2; ++j)
16                temp.mat[i][j] = mat[i][0] * m.mat[0][j] + mat[i][1] * m.mat[1][j];
17        return temp;
18    }
19
20    void display() {
21        for (int i = 0; i < 2; ++i) {
22            for (int j = 0; j < 2; ++j)
23                cout << mat[i][j] << " ";
24            cout << endl;
25        }
26    }
27 };
28
29 int main() {
30     Matrix m1(1, 2, 3, 4), m2(2, 0, 1, 2);
31     Matrix m3 = m1 * m2;
32     m3.display();
33     return 0;
34 }
```



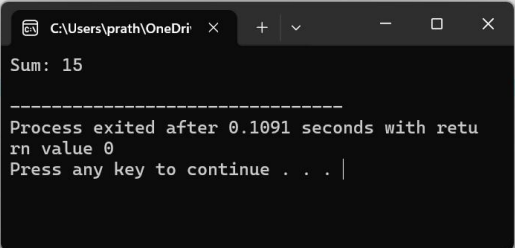
6. Write a c++ program to overload the [] operator to access the elements in an array using index values

```
6. [] operator to access the elements in an array using index values.cpp
1 #include <iostream>
2 using namespace std;
3
4 class Array {
5     int arr[5];
6 public:
7     Array() {
8         for (int i = 0; i < 5; ++i) arr[i] = i * 10;
9     }
10
11     int operator[](int index) {
12         return arr[index];
13     }
14 };
15
16 int main() {
17     Array obj;
18     cout << "Element at index 2: " << obj[2] << endl;
19     return 0;
20 }
21
```



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

```
7. () to call a function with arguments.cpp
1 #include <iostream>
2 using namespace std;
3
4 class FunctionCall {
5 public:
6     void operator()(int a, int b) {
7         cout << "Sum: " << a + b << endl;
8     }
9 };
10
11 int main() {
12     FunctionCall obj;
13     obj(5, 10);
14     return 0;
15 }
16
```



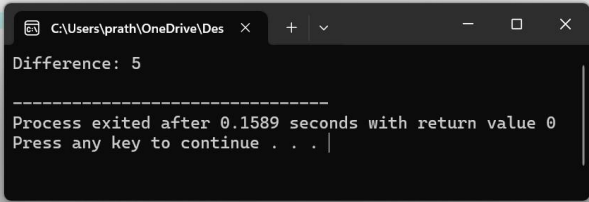
Sum: 15

Process exited after 0.1091 seconds with return value 0
Press any key to continue . . .

8. Write a c++ program to overload the - operator to subtract two variables

```
8. - operator to subtract two variables.cpp
1 #include <iostream>
2 using namespace std;
3
4 class Subtract {
5     int value;
6 public:
7     Subtract(int v) : value(v) {}
8
9     Subtract operator-(const Subtract &obj) {
10         return Subtract(value - obj.value);
11     }
12
13     void display() {
14         cout << "Difference: " << value << endl;
15     }
16 }
17
18 int main() {
19     Subtract obj1(10), obj2(5);
20     Subtract obj3 = obj1 - obj2;
21     obj3.display();
22     return 0;
23 }

```

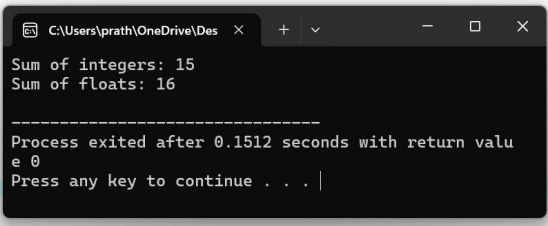


Difference: 5

Process exited after 0.1589 seconds with return value 0
Press any key to continue . . .

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

```
9. function to add two integer numbers and two floating point number separately.cpp
1 #include <iostream>
2 using namespace std;
3
4 class Add {
5 public:
6     int add(int a, int b) {
7         return a + b;
8     }
9
10    float add(float a, float b) {
11        return a + b;
12    }
13 };
14
15 int main() {
16     Add obj;
17     cout << "Sum of integers: " << obj.add(5, 10) << endl;
18     cout << "Sum of floats: " << obj.add(5.5f, 10.5f) << endl;
19     return 0;
20 }
21
```

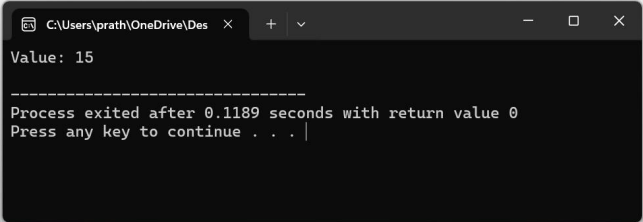


Sum of integers: 15
Sum of floats: 16

Process exited after 0.1512 seconds with return value 0
Press any key to continue . . .

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

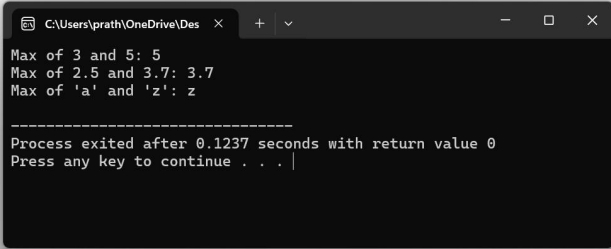
```
10. += operator to add two objects of a user defined class.cpp
1 #include <iostream>
2 using namespace std;
3
4 class AddAssign {
5     int value;
6 public:
7     AddAssign(int v) : value(v) {}
8
9     void operator+=(const AddAssign &obj) {
10         value += obj.value;
11     }
12
13     void display() {
14         cout << "Value: " << value << endl;
15     }
16 };
17
18 int main() {
19     AddAssign obj1(5), obj2(10);
20     obj1 += obj2;
21     obj1.display();
22     return 0;
23 }
24
```



```
C:\Users\prath\OneDrive\Des
Value: 15
-----
Process exited after 0.1189 seconds with return value 0
Press any key to continue . . .
```

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

```
11. find the maximum value from two integer numbers and two floating point number, and two characters separately.cpp
1 #include <iostream>
2 using namespace std;
3
4 int max(int a, int b) {
5     return (a > b) ? a : b;
6 }
7
8 float max(float a, float b) {
9     return (a > b) ? a : b;
10 }
11
12 char max(char a, char b) {
13     return (a > b) ? a : b;
14 }
15
16 int main() {
17     cout << "Max of 3 and 5: " << max(3, 5) << endl;
18     cout << "Max of 2.5 and 3.7: " << max(2.5f, 3.7f) << endl;
19     cout << "Max of 'a' and 'z': " << max('a', 'z') << endl;
20     return 0;
21 }
22
```



```
C:\Users\prath\OneDrive\Des
Max of 3 and 5: 5
Max of 2.5 and 3.7: 3.7
Max of 'a' and 'z': z
-----
Process exited after 0.1237 seconds with return value 0
Press any key to continue . . .
```

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

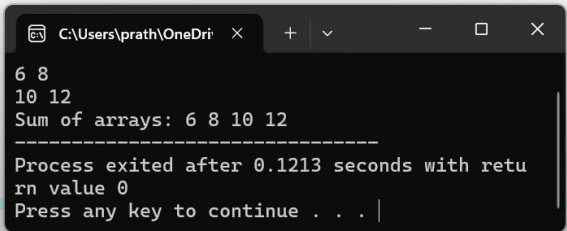
```
12. concatenate two strings and two characters arrays separately.cpp
1 #include <iostream>
2 #include <cstring>
3 using namespace std;
4
5 string concatenate(string a, string b) {
6     return a + b;
7 }
8
9 void concatenate(char a[], char b[], char result[]) {
10     strcpy(result, a);
11     strcat(result, b);
12 }
13
14 int main() {
15     string str1 = "Hello", str2 = "World";
16     cout << "Concatenated string: " << concatenate(str1, str2) << endl;
17
18     char arr1[] = "Hello", arr2[] = "World", result[100];
19     concatenate(arr1, arr2, result);
20     cout << "Concatenated char arrays: " << result << endl;
21
22     return 0;
23 }
24
```



```
C:\Users\prath\OneDrive\Des
Concatenated string: HelloWorld
Concatenated char arrays: HelloWorld
-----
Process exited after 0.1073 seconds with return value 0
Press any key to continue . . .
```

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

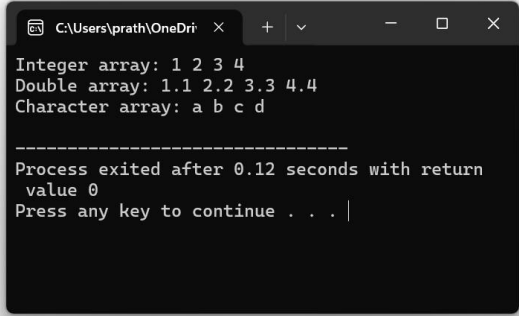
```
13. calculate the sum of two matrices and two arrays separately.cpp
1 #include <iostream>
2 using namespace std;
3 void add(int mat1[2][2], int mat2[2][2], int result[2][2]) {
4     for (int i = 0; i < 2; i++) {
5         for (int j = 0; j < 2; j++) {
6             result[i][j] = mat1[i][j] + mat2[i][j];
7         }
8     }
9 }
10 void add(int arr1[], int arr2[], int result[], int size) {
11     for (int i = 0; i < size; i++) {
12         result[i] = arr1[i] + arr2[i];
13     }
14 }
15 int main() {
16     int mat1[2][2] = {{1, 2}, {3, 4}};
17     int mat2[2][2] = {{5, 6}, {7, 8}};
18     int resultMatrix[2][2];
19     add(mat1, mat2, resultMatrix);
20     cout << "Sum of matrices:\n";
21     for (int i = 0; i < 2; i++) {
22         for (int j = 0; j < 2; j++) {
23             cout << resultMatrix[i][j] << " ";
24         }
25         cout << endl;
26     }
27     int arr1[] = {1, 2, 3, 4};
28     int arr2[] = {5, 6, 7, 8};
29     int resultArray[4];
30     add(arr1, arr2, resultArray, 4);
31     cout << "Sum of arrays: ";
32     for (int i = 0; i < 4; i++) {
33         cout << resultArray[i] << " ";
34     }
35     return 0;
36 }
```



```
C:\Users\prath\OneDrive\
6 8
10 12
Sum of arrays: 6 8 10 12
-----
Process exited after 0.1213 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

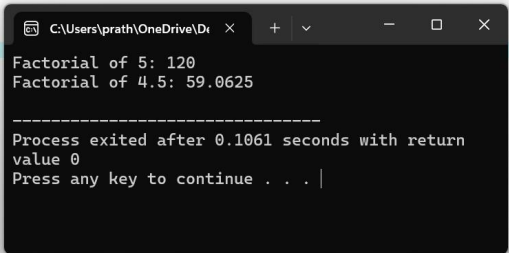
```
14. print an integer array, a double array and a character array separately.cpp
1 #include <iostream>
2 using namespace std;
3 void printArray(int arr[], int size) {
4     for (int i = 0; i < size; i++) {
5         cout << arr[i] << " ";
6     }
7     cout << endl;
8 }
9 void printArray(double arr[], int size) {
10     for (int i = 0; i < size; i++) {
11         cout << arr[i] << " ";
12     }
13     cout << endl;
14 }
15 void printArray(char arr[], int size) {
16     for (int i = 0; i < size; i++) {
17         cout << arr[i] << " ";
18     }
19     cout << endl;
20 }
21 int main() {
22     int intArr[] = {1, 2, 3, 4};
23     double doubleArr[] = {1.1, 2.2, 3.3, 4.4};
24     char charArr[] = {'a', 'b', 'c', 'd'};
25     cout << "Integer array: ";
26     printArray(intArr, 4);
27     cout << "Double array: ";
28     printArray(doubleArr, 4);
29     cout << "Character array: ";
30     printArray(charArr, 4);
31     return 0;
32 }
```



```
C:\Users\prath\OneDrive\
Integer array: 1 2 3 4
Double array: 1.1 2.2 3.3 4.4
Character array: a b c d
-----
Process exited after 0.12 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

```
15. factorial of an integer number and factorial of a floating-point number separately.cpp
1 #include <iostream>
2 using namespace std;
3 int factorial(int n) {
4     if (n <= 1) return 1;
5     return n * factorial(n - 1);
6 }
7 float factorial(float n) {
8     if (n <= 1) return 1;
9     return n * factorial(n - 1);
10 }
11 int main() {
12     int intNum = 5;
13     float floatNum = 4.5f;
14     cout << "Factorial of " << intNum << ": " << factorial(intNum) << endl;
15     cout << "Factorial of " << floatNum << ": " << factorial(floatNum) << endl;
16     return 0;
17 }
```



```
C:\Users\prath\OneDrive\
Factorial of 5: 120
Factorial of 4.5: 59.0625
-----
Process exited after 0.1061 seconds with return value 0
Press any key to continue . . .
```

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

```
16. function to sort an integer array and a double array.cpp
1  #include <iostream>
2  #include <algorithm>
3  using namespace std;
4  void sortArray(int arr[], int size) {
5      sort(arr, arr + size);
6  }
7  void sortArray(double arr[], int size) {
8      sort(arr, arr + size);
9  }
10 int main() {
11     int intArr[] = {4, 2, 3, 1};
12     double doubleArr[] = {4.4, 2.2, 3.3, 1.1};
13     sortArray(intArr, 4);
14     sortArray(doubleArr, 4);
15     cout << "Sorted integer array: ";
16     for (int i = 0; i < 4; i++) {
17         cout << intArr[i] << " ";
18     }
19     cout << endl;
20     cout << "Sorted double array: ";
21     for (int i = 0; i < 4; i++) {
22         cout << doubleArr[i] << " ";
23     }
24     return 0;
25 }
```

```
C:\Users\prath\OneDrive\Des
Sorted integer array: 1 2 3 4
Sorted double array: 1.1 2.2
-----
Process exited after 0.1136 seconds with return value 0
Press any key to continue . . .
```

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

```
17. calculate the power of an integer number and power of a floating-point number separately.cpp
1  #include <iostream>
2  #include <cmath>
3  using namespace std;
4
5  int power(int base, int exp) {
6      return pow(base, exp);
7  }
8
9  float power(float base, float exp) {
10     return pow(base, exp);
11 }
12
13 int main() {
14     int intBase = 2, intExp = 3;
15     float floatBase = 2.5f, floatExp = 3.5f;
16
17     cout << intBase << " raised to " << intExp << " is " << power(intBase, intExp) << endl;
18     cout << floatBase << " raised to " << floatExp << " is " << power(floatBase, floatExp) << endl;
19
20     return 0;
21 }
```

```
C:\Users\prath\OneDrive\Des
2 raised to 3 is 8
2.5 raised to 3.5 is 24.7053
-----
Process exited after 0.111 seconds with return 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

```
18. find an absolute value of an integer number and absolute value of a floating-point number separately.cpp
1  #include <iostream>
2  using namespace std;
3
4  int absolute(int n) {
5      return (n < 0) ? -n : n;
6  }
7
8  float absolute(float n) {
9      return (n < 0) ? -n : n;
10 }
11
12 int main() {
13     int intNum = -5;
14     float floatNum = -3.5f;
15
16     cout << "Absolute value of " << intNum << " is " << absolute(intNum) << endl;
17     cout << "Absolute value of " << floatNum << " is " << absolute(floatNum) << endl;
18
19     return 0;
20 }
21
```

```
C:\Users\prath\OneDrive\Des
Absolute value of -5 is 5
Absolute value of -3.5 is 3.5
-----
Process exited after 0.1144 seconds with return value 0
Press any key to continue . . .
```