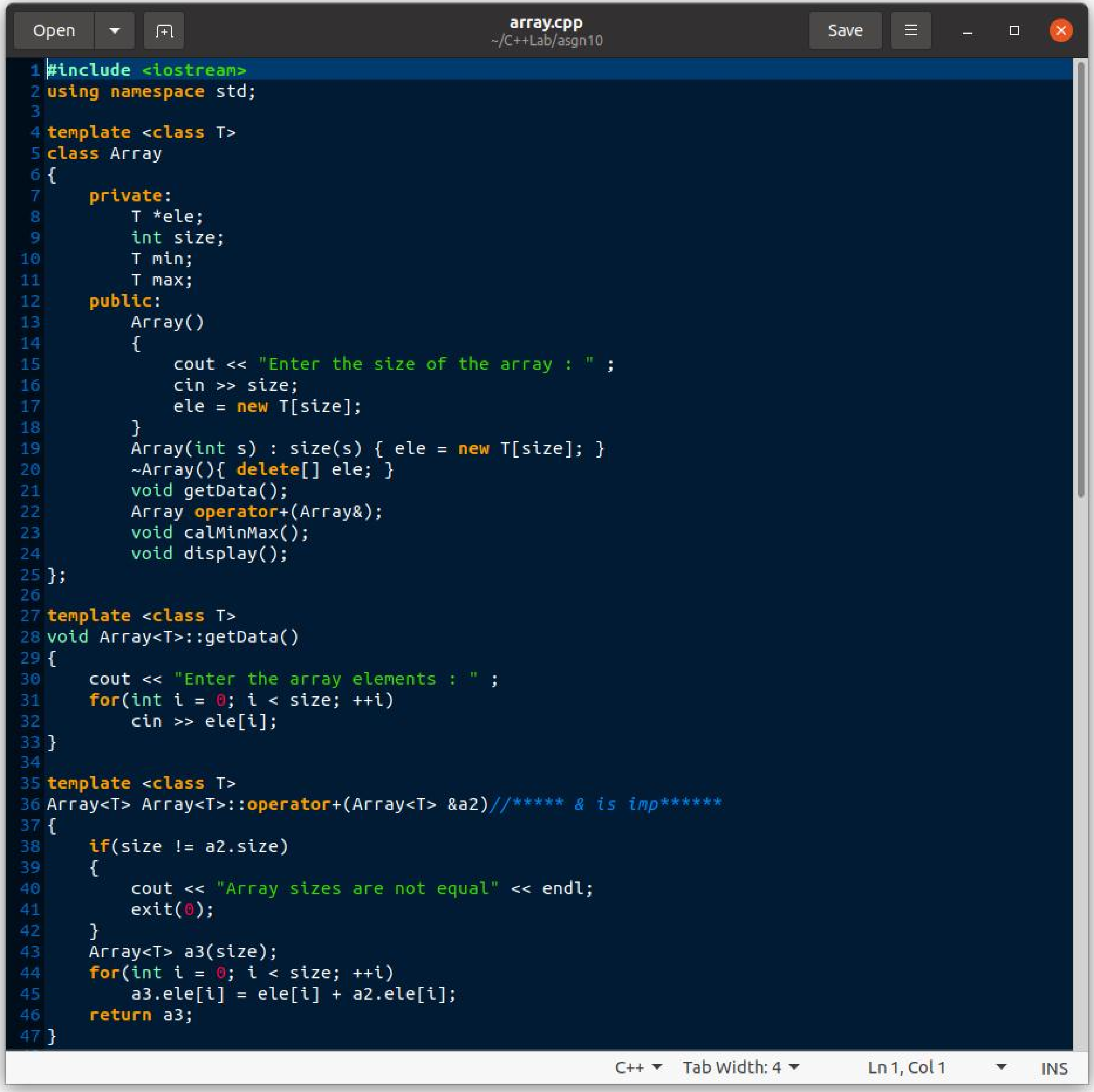
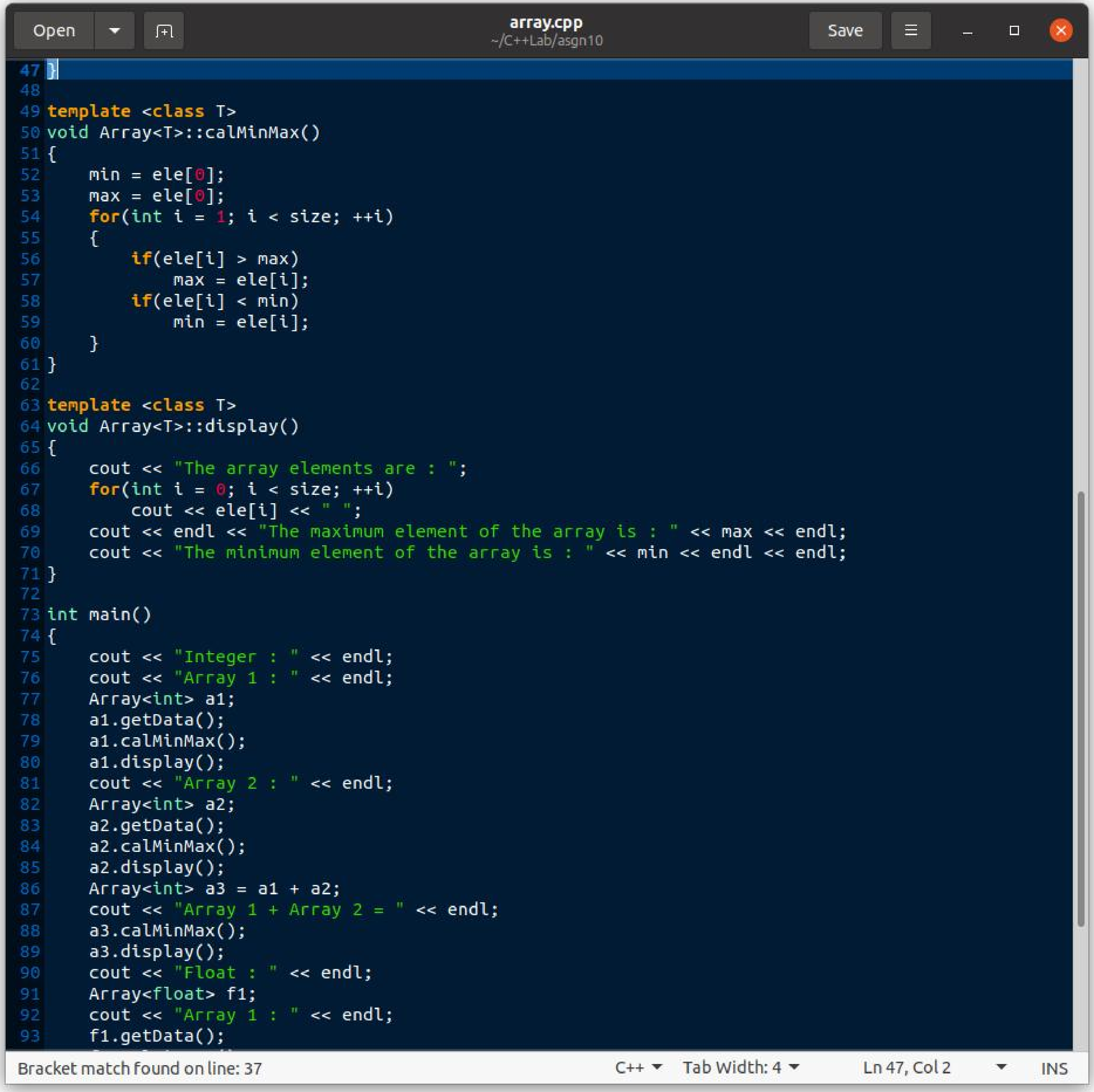
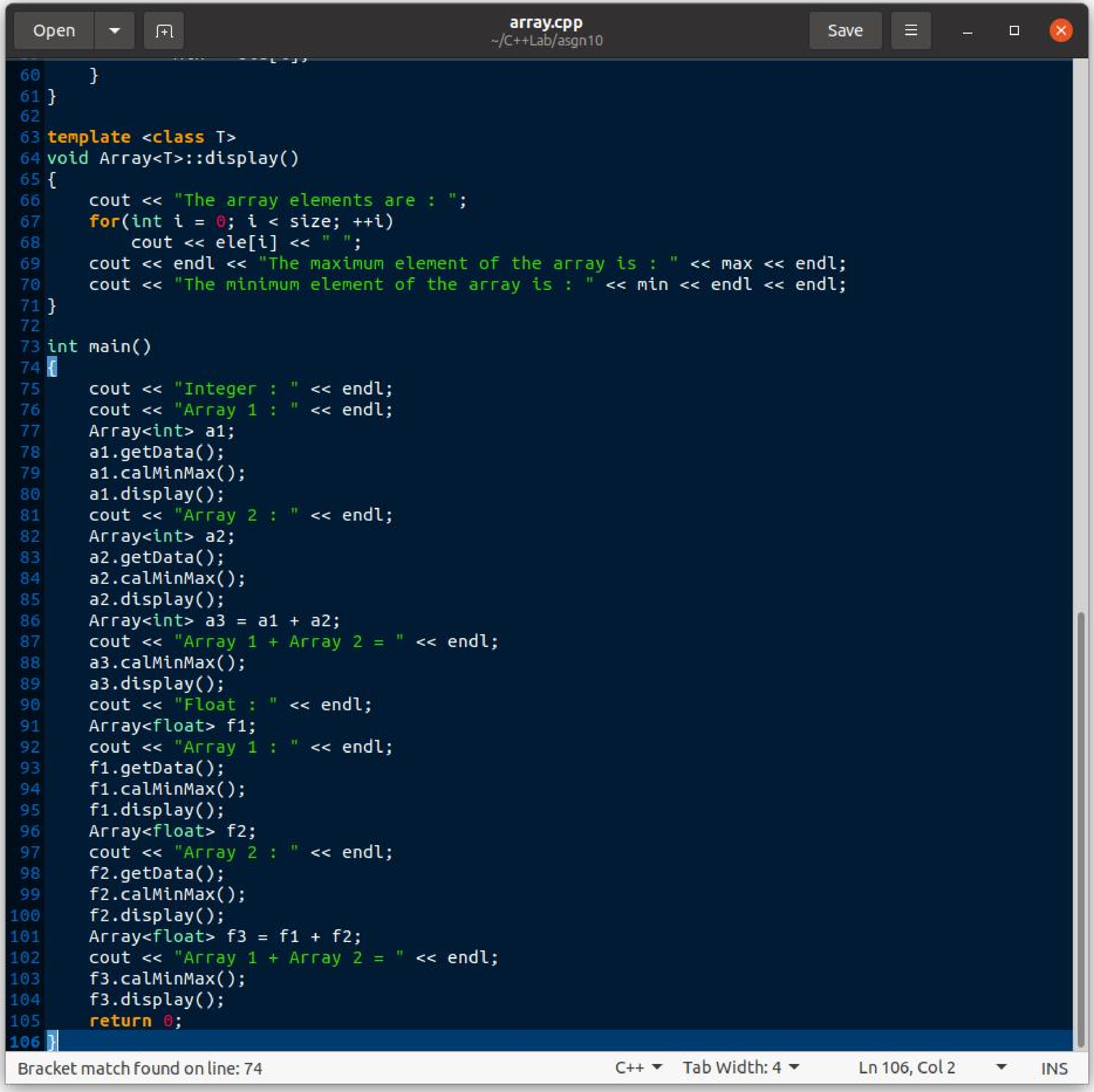


```cpp
#include <iostream>
using namespace std;

template <class T>
class Array
{
    private:
        T *ele;
        int size;
        T min;
        T max;
    public:
        Array()
        {
            cout << "Enter the size of the array : " ;
            cin >> size;
            ele = new T[size];
        }
        Array(int s) : size(s) { ele = new T[size]; }
        ~Array(){ delete[] ele; }
        void getData();
        Array operator+(Array&);
        void calMinMax();
        void display();
};

template <class T>
void Array<T>::getData()
{
    cout << "Enter the array elements : " ;
    for(int i = 0; i < size; ++i)
        cin >> ele[i];
}

template <class T>
Array<T> Array<T>::operator+(Array<T> &a2)//***** & is imp******
{
    if(size != a2.size)
    {
        cout << "Array sizes are not equal" << endl;
        exit(0);
    }
    Array<T> a3(size);
    for(int i = 0; i < size; ++i)
        a3.ele[i] = ele[i] + a2.ele[i];
    return a3;
}
```



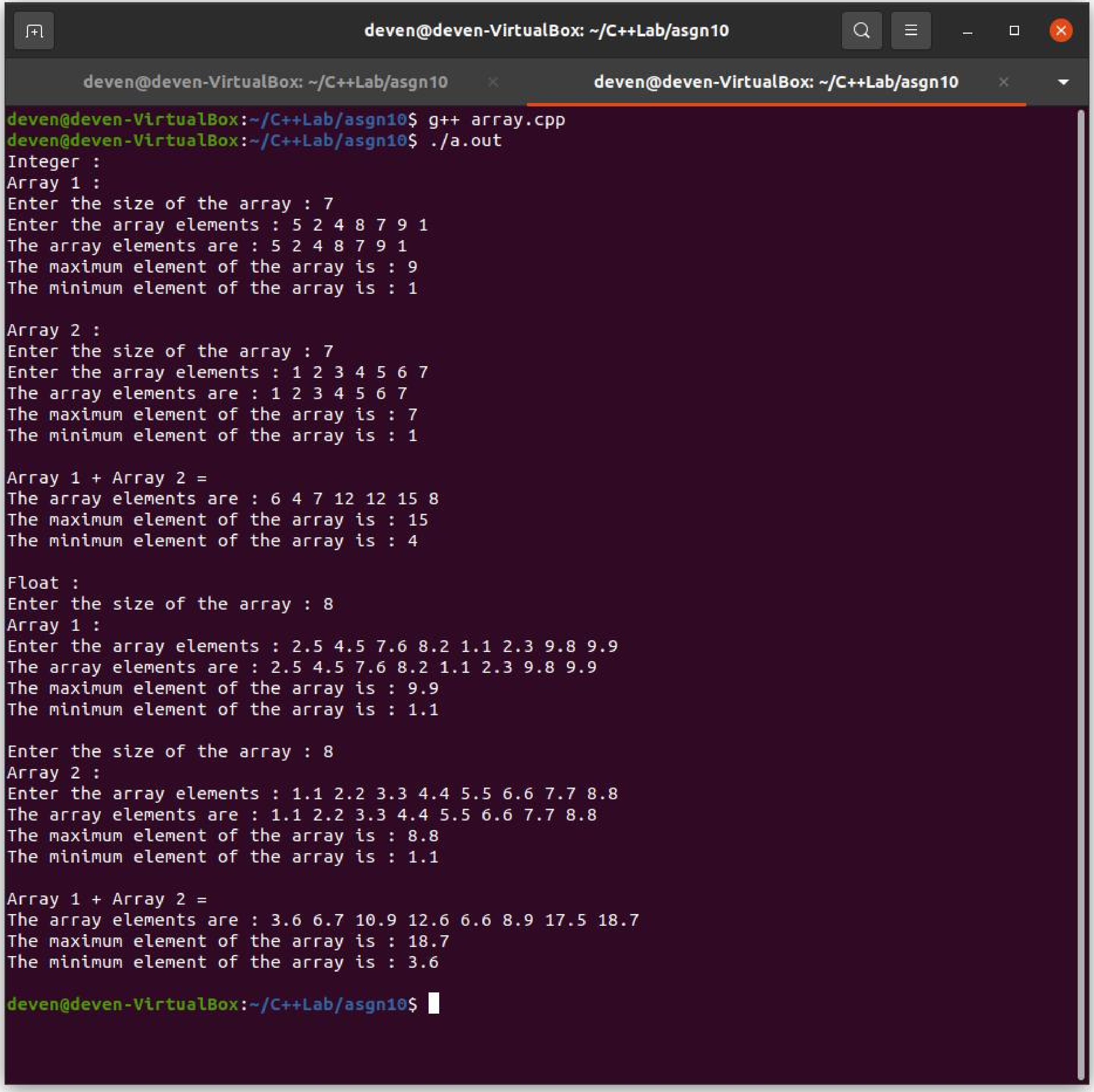
```cpp
}

template <class T>
void Array<T>::calMinMax()
{
    min = ele[0];
    max = ele[0];
    for(int i = 1; i < size; ++i)
    {
        if(ele[i] > max)
            max = ele[i];
        if(ele[i] < min)
            min = ele[i];
    }
}

template <class T>
void Array<T>::display()
{
    cout << "The array elements are : ";
    for(int i = 0; i < size; ++i)
        cout << ele[i] << " ";
    cout << endl << "The maximum element of the array is : " << max << endl;
    cout << "The minimum element of the array is : " << min << endl << endl;
}

int main()
{
    cout << "Integer : " << endl;
    cout << "Array 1 : " << endl;
    Array<int> a1;
    a1.getData();
    a1.calMinMax();
    a1.display();
    cout << "Array 2 : " << endl;
    Array<int> a2;
    a2.getData();
    a2.calMinMax();
    a2.display();
    Array<int> a3 = a1 + a2;
    cout << "Array 1 + Array 2 = " << endl;
    a3.calMinMax();
    a3.display();
    cout << "Float : " << endl;
    Array<float> f1;
    cout << "Array 1 : " << endl;
    f1.getData();
```



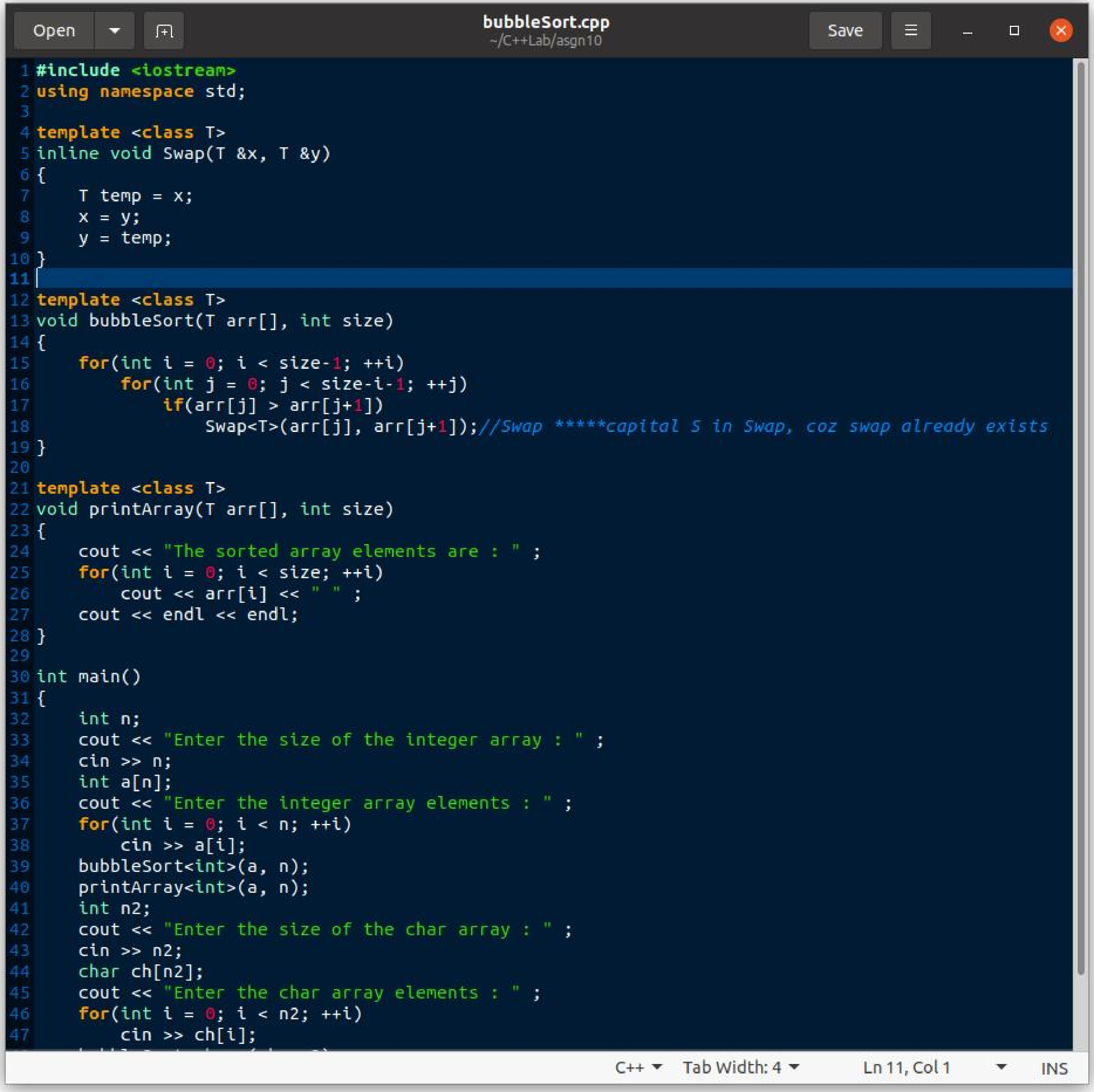
array.cpp
~/C++Lab/asgn10

```cpp
    }
}

template <class T>
void Array<T>::display()
{
    cout << "The array elements are : ";
    for(int i = 0; i < size; ++i)
        cout << ele[i] << " ";
    cout << endl << "The maximum element of the array is : " << max << endl;
    cout << "The minimum element of the array is : " << min << endl << endl;
}

int main()
{
    cout << "Integer : " << endl;
    cout << "Array 1 : " << endl;
    Array<int> a1;
    a1.getData();
    a1.calMinMax();
    a1.display();
    cout << "Array 2 : " << endl;
    Array<int> a2;
    a2.getData();
    a2.calMinMax();
    a2.display();
    Array<int> a3 = a1 + a2;
    cout << "Array 1 + Array 2 = " << endl;
    a3.calMinMax();
    a3.display();
    cout << "Float : " << endl;
    Array<float> f1;
    cout << "Array 1 : " << endl;
    f1.getData();
    f1.calMinMax();
    f1.display();
    Array<float> f2;
    cout << "Array 2 : " << endl;
    f2.getData();
    f2.calMinMax();
    f2.display();
    Array<float> f3 = f1 + f2;
    cout << "Array 1 + Array 2 = " << endl;
    f3.calMinMax();
    f3.display();
    return 0;
}
```

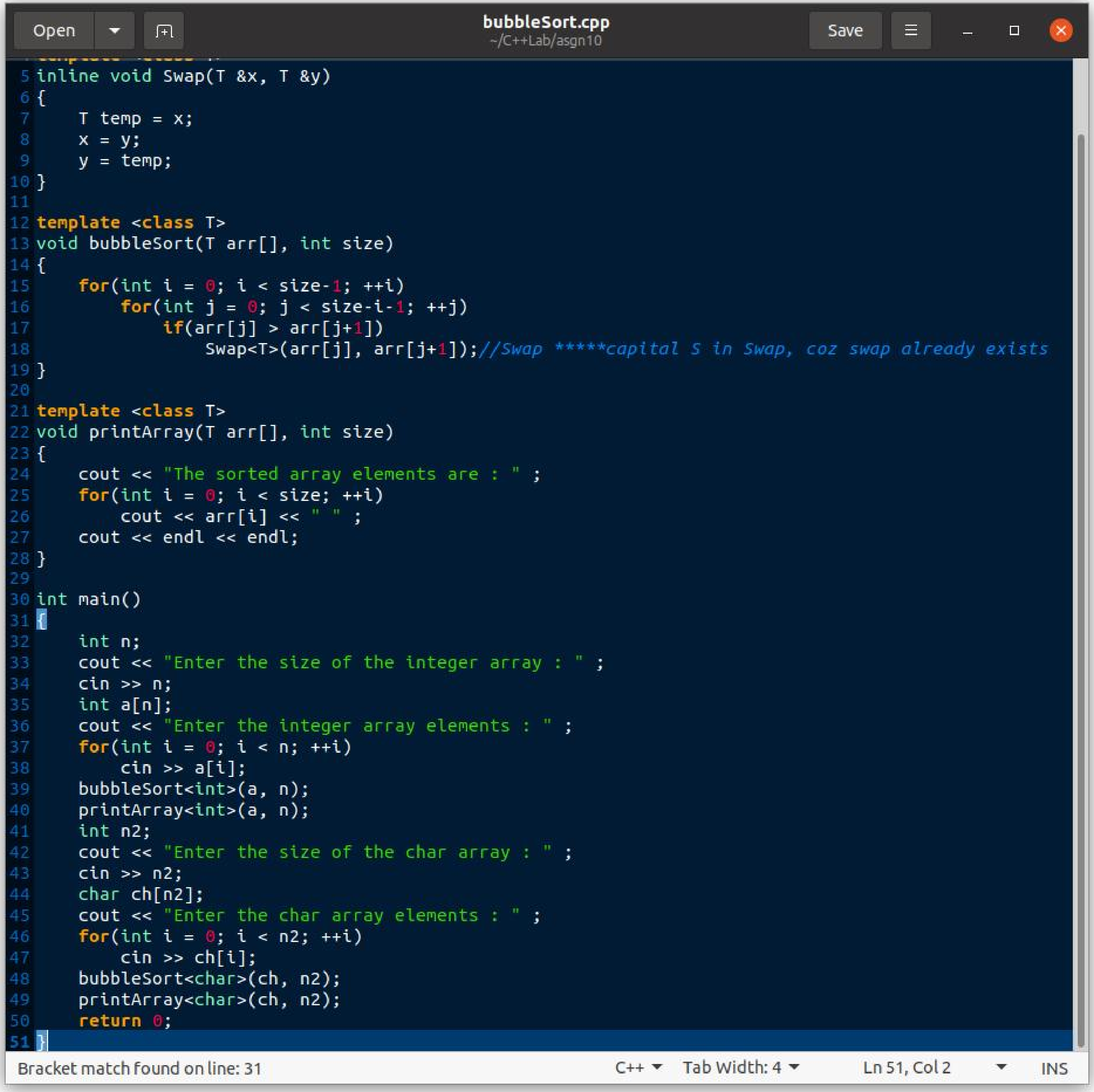
Bracket match found on line: 74   C++   Tab Width: 4   Ln 106, Col 2   INS


deven@deven-VirtualBox: ~/C++Lab/asgn10
deven@deven-VirtualBox: ~/C++Lab/asgn10
deven@deven-VirtualBox: ~/C++Lab/asgn10

```
deven@deven-VirtualBox:~/C++Lab/asgn10$ g++ array.cpp
deven@deven-VirtualBox:~/C++Lab/asgn10$ ./a.out
Integer :
Array 1 :
Enter the size of the array : 7
Enter the array elements : 5 2 4 8 7 9 1
The array elements are : 5 2 4 8 7 9 1
The maximum element of the array is : 9
The minimum element of the array is : 1

Array 2 :
Enter the size of the array : 7
Enter the array elements : 1 2 3 4 5 6 7
The array elements are : 1 2 3 4 5 6 7
The maximum element of the array is : 7
The minimum element of the array is : 1

Array 1 + Array 2 =
The array elements are : 6 4 7 12 12 15 8
The maximum element of the array is : 15
The minimum element of the array is : 4

Float :
Enter the size of the array : 8
Array 1 :
Enter the array elements : 2.5 4.5 7.6 8.2 1.1 2.3 9.8 9.9
The array elements are : 2.5 4.5 7.6 8.2 1.1 2.3 9.8 9.9
The maximum element of the array is : 9.9
The minimum element of the array is : 1.1

Enter the size of the array : 8
Array 2 :
Enter the array elements : 1.1 2.2 3.3 4.4 5.5 6.6 7.7 8.8
The array elements are : 1.1 2.2 3.3 4.4 5.5 6.6 7.7 8.8
The maximum element of the array is : 8.8
The minimum element of the array is : 1.1

Array 1 + Array 2 =
The array elements are : 3.6 6.7 10.9 12.6 6.6 8.9 17.5 18.7
The maximum element of the array is : 18.7
The minimum element of the array is : 3.6

deven@deven-VirtualBox:~/C++Lab/asgn10$
```



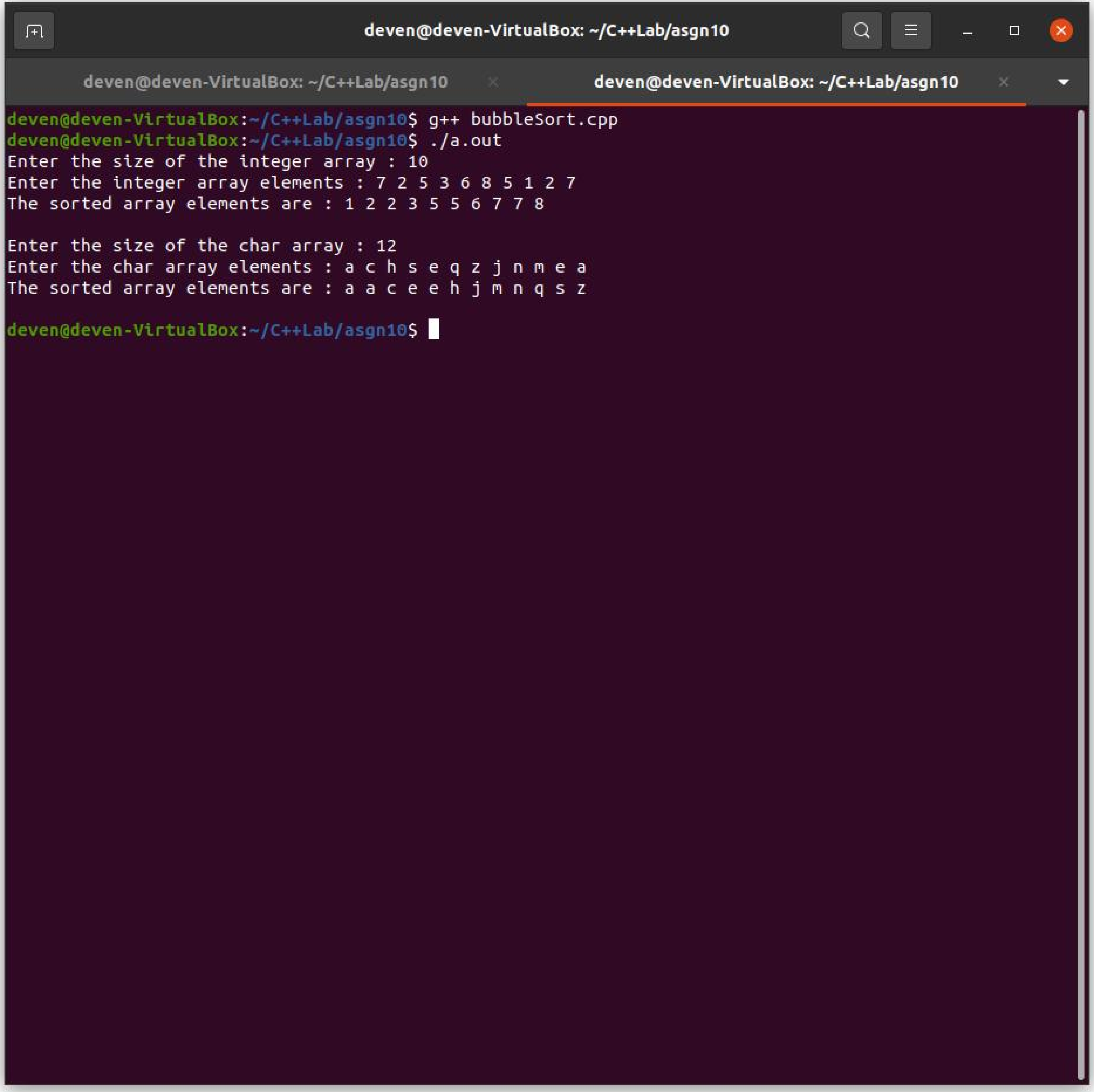
```cpp
#include <iostream>
using namespace std;

template <class T>
inline void Swap(T &x, T &y)
{
    T temp = x;
    x = y;
    y = temp;
}

template <class T>
void bubbleSort(T arr[], int size)
{
    for(int i = 0; i < size-1; ++i)
        for(int j = 0; j < size-i-1; ++j)
            if(arr[j] > arr[j+1])
                Swap<T>(arr[j], arr[j+1]);//Swap *****capital S in Swap, coz swap already exists
}

template <class T>
void printArray(T arr[], int size)
{
    cout << "The sorted array elements are : " ;
    for(int i = 0; i < size; ++i)
        cout << arr[i] << " " ;
    cout << endl << endl;
}

int main()
{
    int n;
    cout << "Enter the size of the integer array : " ;
    cin >> n;
    int a[n];
    cout << "Enter the integer array elements : " ;
    for(int i = 0; i < n; ++i)
        cin >> a[i];
    bubbleSort<int>(a, n);
    printArray<int>(a, n);
    int n2;
    cout << "Enter the size of the char array : " ;
    cin >> n2;
    char ch[n2];
    cout << "Enter the char array elements : " ;
    for(int i = 0; i < n2; ++i)
        cin >> ch[i];
```



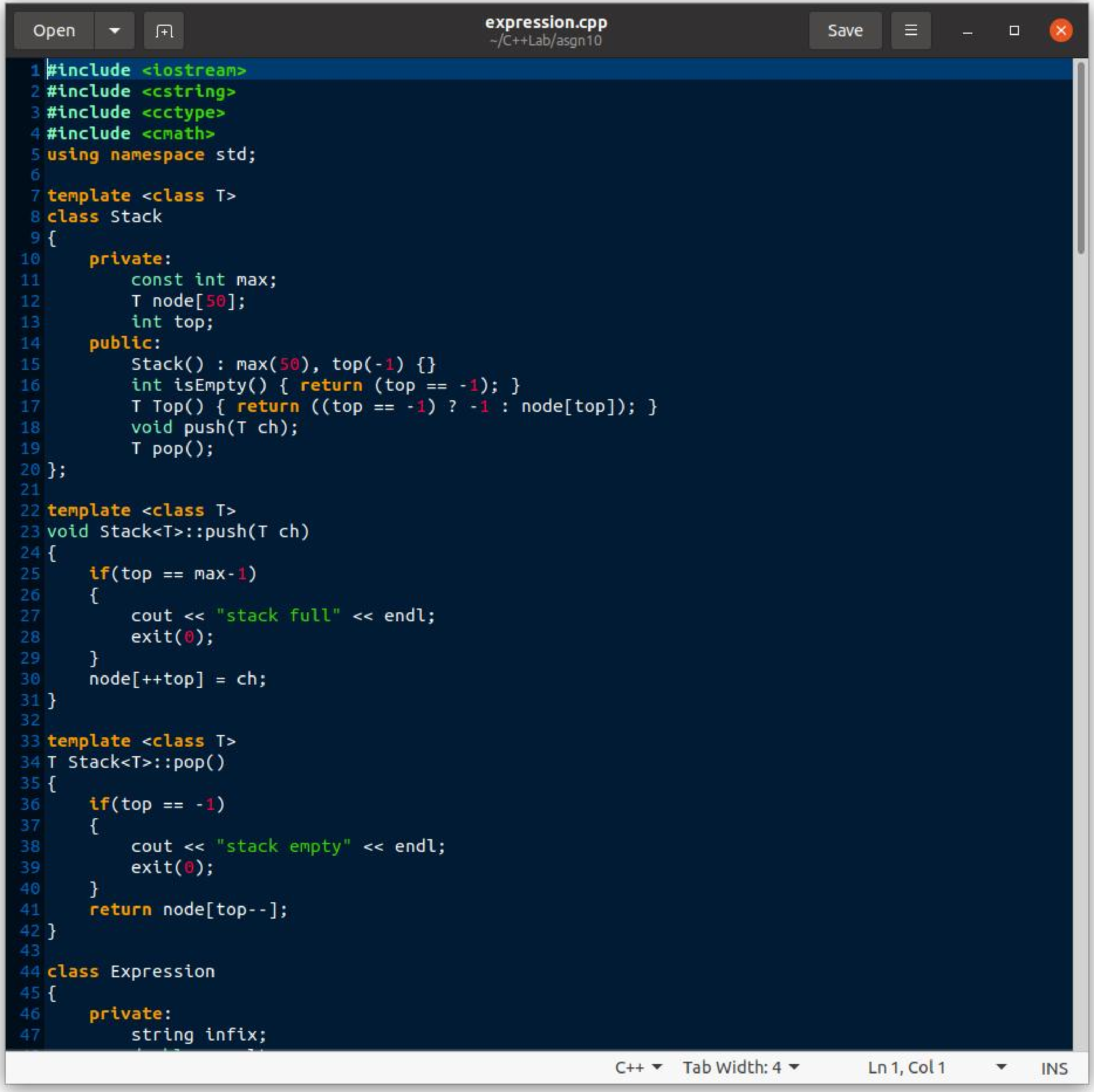
```cpp
template <class T>
inline void Swap(T &x, T &y)
{
    T temp = x;
    x = y;
    y = temp;
}

template <class T>
void bubbleSort(T arr[], int size)
{
    for(int i = 0; i < size-1; ++i)
        for(int j = 0; j < size-i-1; ++j)
            if(arr[j] > arr[j+1])
                Swap<T>(arr[j], arr[j+1]);//Swap *****capital S in Swap, coz swap already exists
}

template <class T>
void printArray(T arr[], int size)
{
    cout << "The sorted array elements are : " ;
    for(int i = 0; i < size; ++i)
        cout << arr[i] << " " ;
    cout << endl << endl;
}

int main()
{
    int n;
    cout << "Enter the size of the integer array : " ;
    cin >> n;
    int a[n];
    cout << "Enter the integer array elements : " ;
    for(int i = 0; i < n; ++i)
        cin >> a[i];
    bubbleSort<int>(a, n);
    printArray<int>(a, n);
    int n2;
    cout << "Enter the size of the char array : " ;
    cin >> n2;
    char ch[n2];
    cout << "Enter the char array elements : " ;
    for(int i = 0; i < n2; ++i)
        cin >> ch[i];
    bubbleSort<char>(ch, n2);
    printArray<char>(ch, n2);
    return 0;
}
```

Bracket match found on line: 31

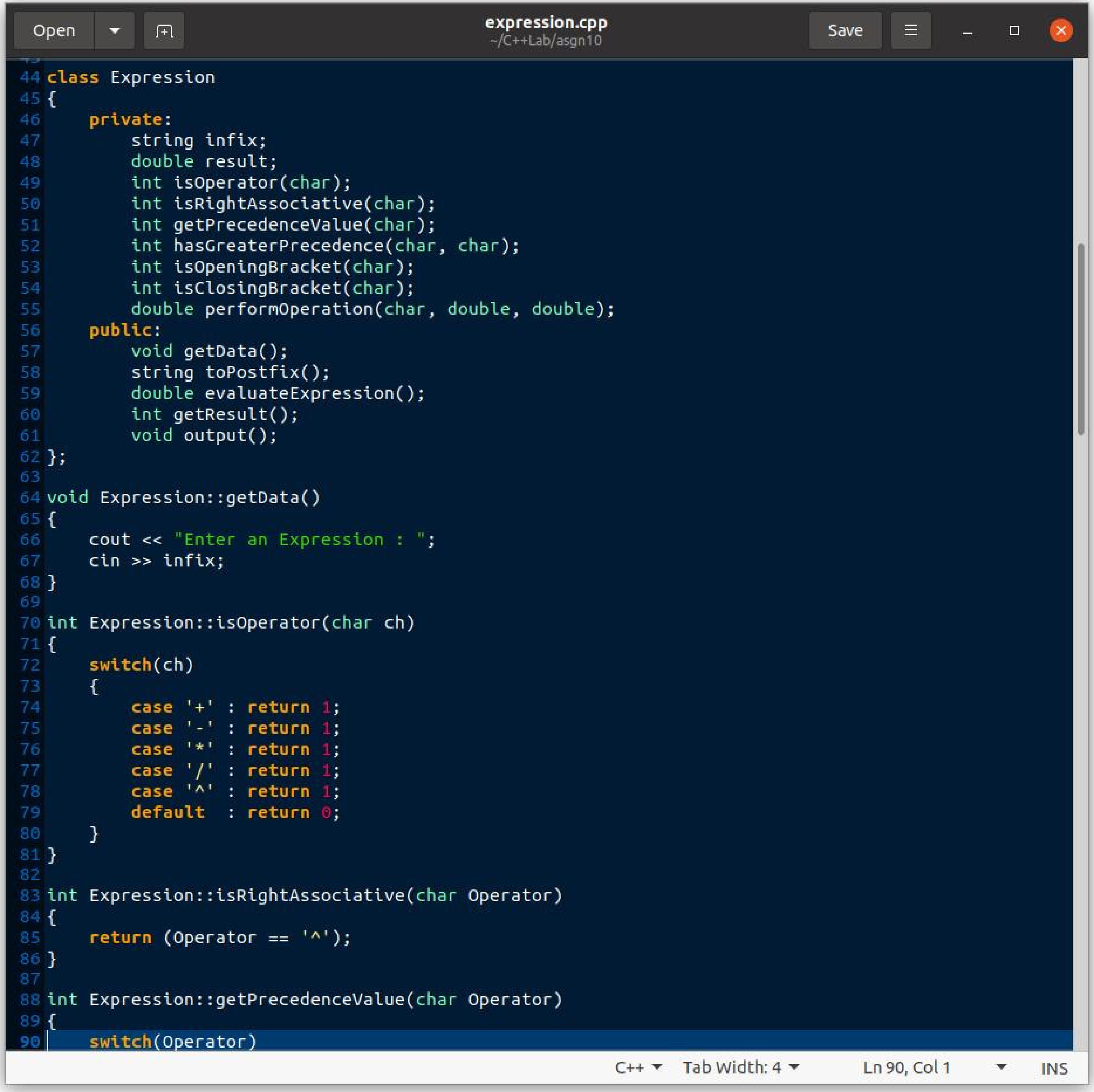
C++ Tab Width: 4 Ln 51, Col 2 INS


deven@deven-VirtualBox: ~/C++Lab/asgn10
deven@deven-VirtualBox: ~/C++Lab/asgn10
deven@deven-VirtualBox: ~/C++Lab/asgn10

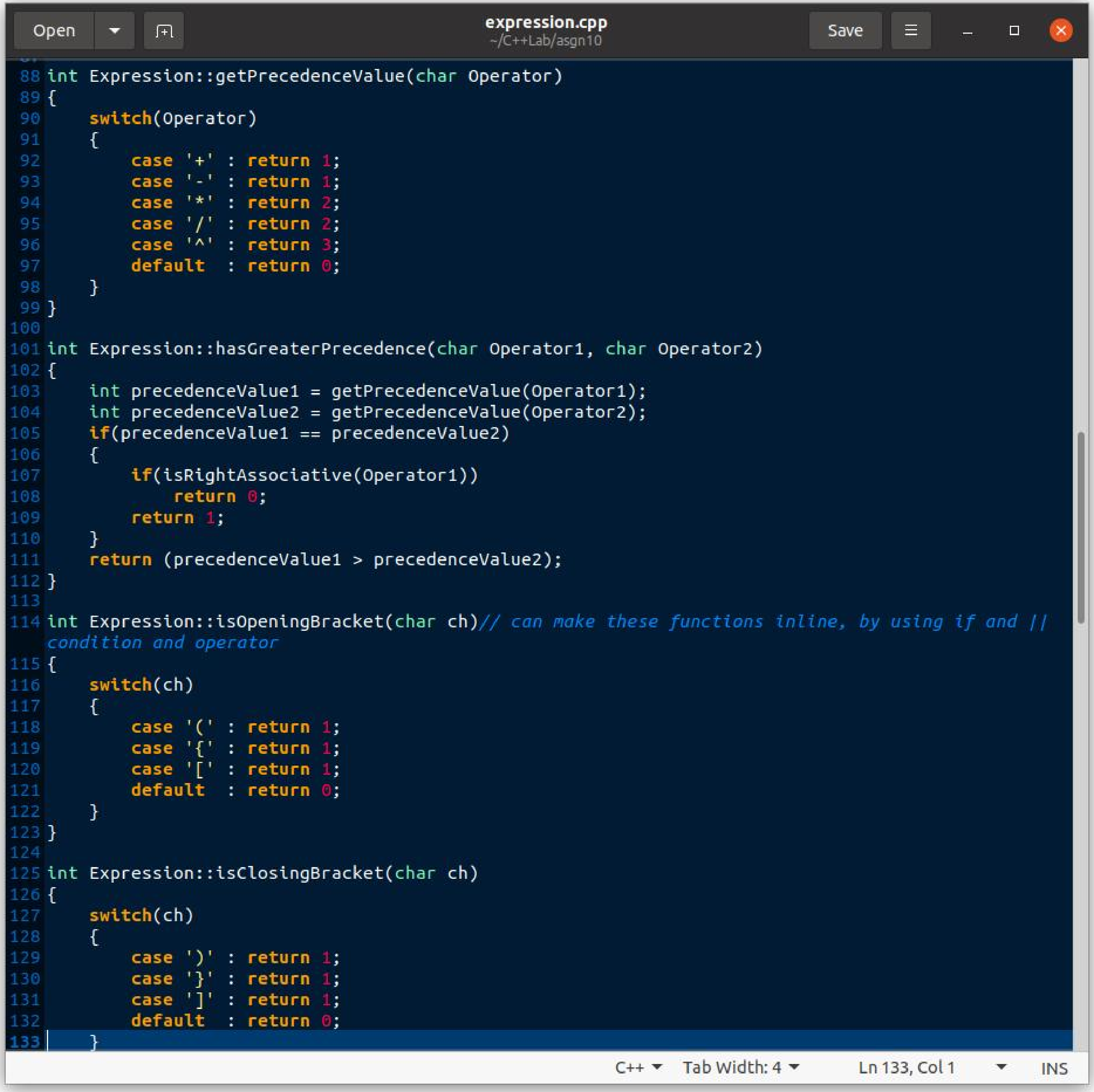
```
deven@deven-VirtualBox:~/C++Lab/asgn10$ g++ bubbleSort.cpp
deven@deven-VirtualBox:~/C++Lab/asgn10$ ./a.out
Enter the size of the integer array : 10
Enter the integer array elements : 7 2 5 3 6 8 5 1 2 7
The sorted array elements are : 1 2 2 3 5 5 6 7 7 8

Enter the size of the char array : 12
Enter the char array elements : a c h s e q z j n m e a
The sorted array elements are : a a c e e h j m n q s z

deven@deven-VirtualBox:~/C++Lab/asgn10$ 
```



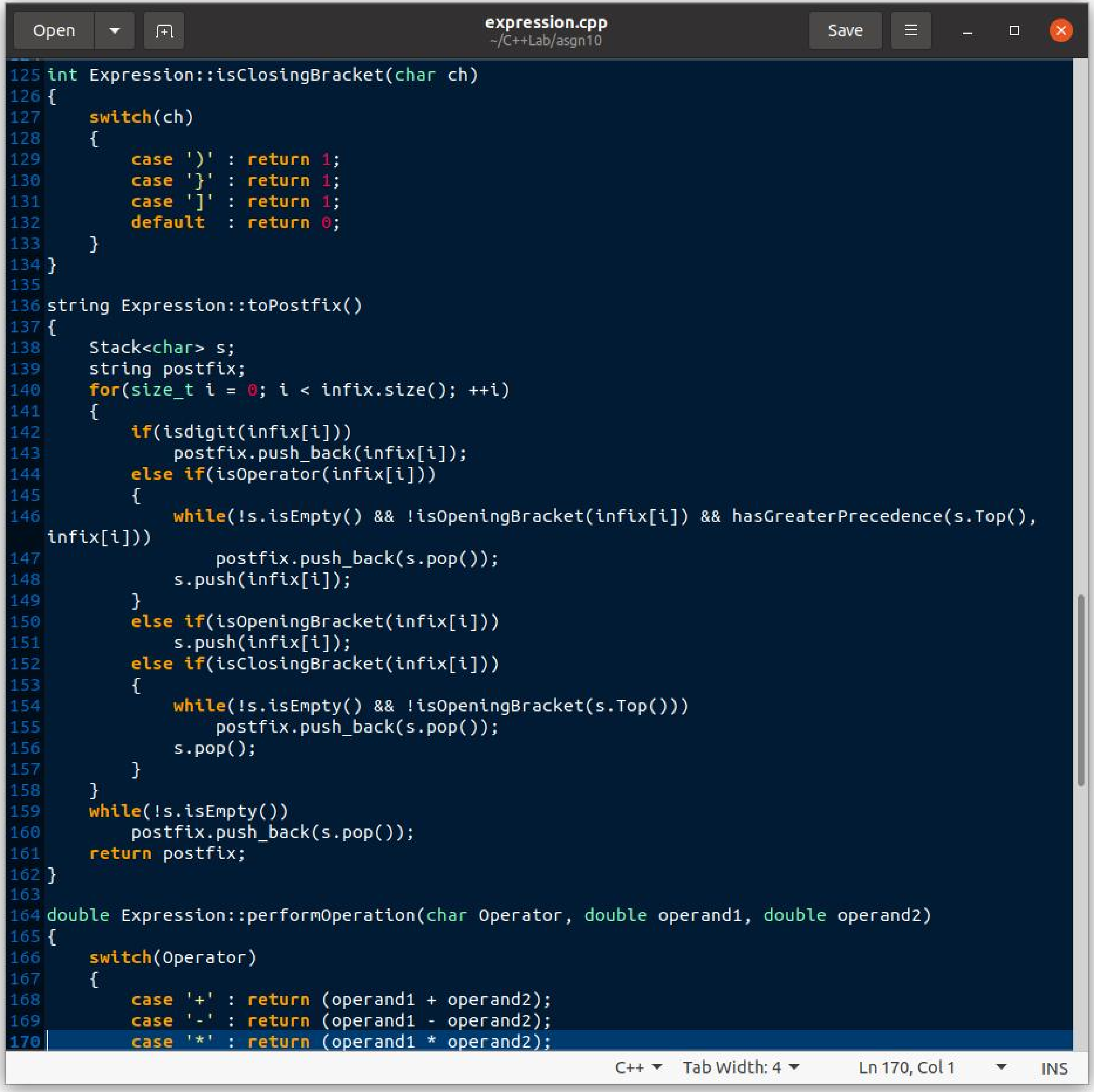
```cpp
#include <iostream>
#include <cstring>
#include <cctype>
#include <cmath>
using namespace std;

template <class T>
class Stack
{
    private:
        const int max;
        T node[50];
        int top;
    public:
        Stack() : max(50), top(-1) {}
        int isEmpty() { return (top == -1); }
        T Top() { return ((top == -1) ? -1 : node[top]); }
        void push(T ch);
        T pop();
};

template <class T>
void Stack<T>::push(T ch)
{
    if(top == max-1)
    {
        cout << "stack full" << endl;
        exit(0);
    }
    node[++top] = ch;
}

template <class T>
T Stack<T>::pop()
{
    if(top == -1)
    {
        cout << "stack empty" << endl;
        exit(0);
    }
    return node[top--];
}

class Expression
{
    private:
        string infix;
```



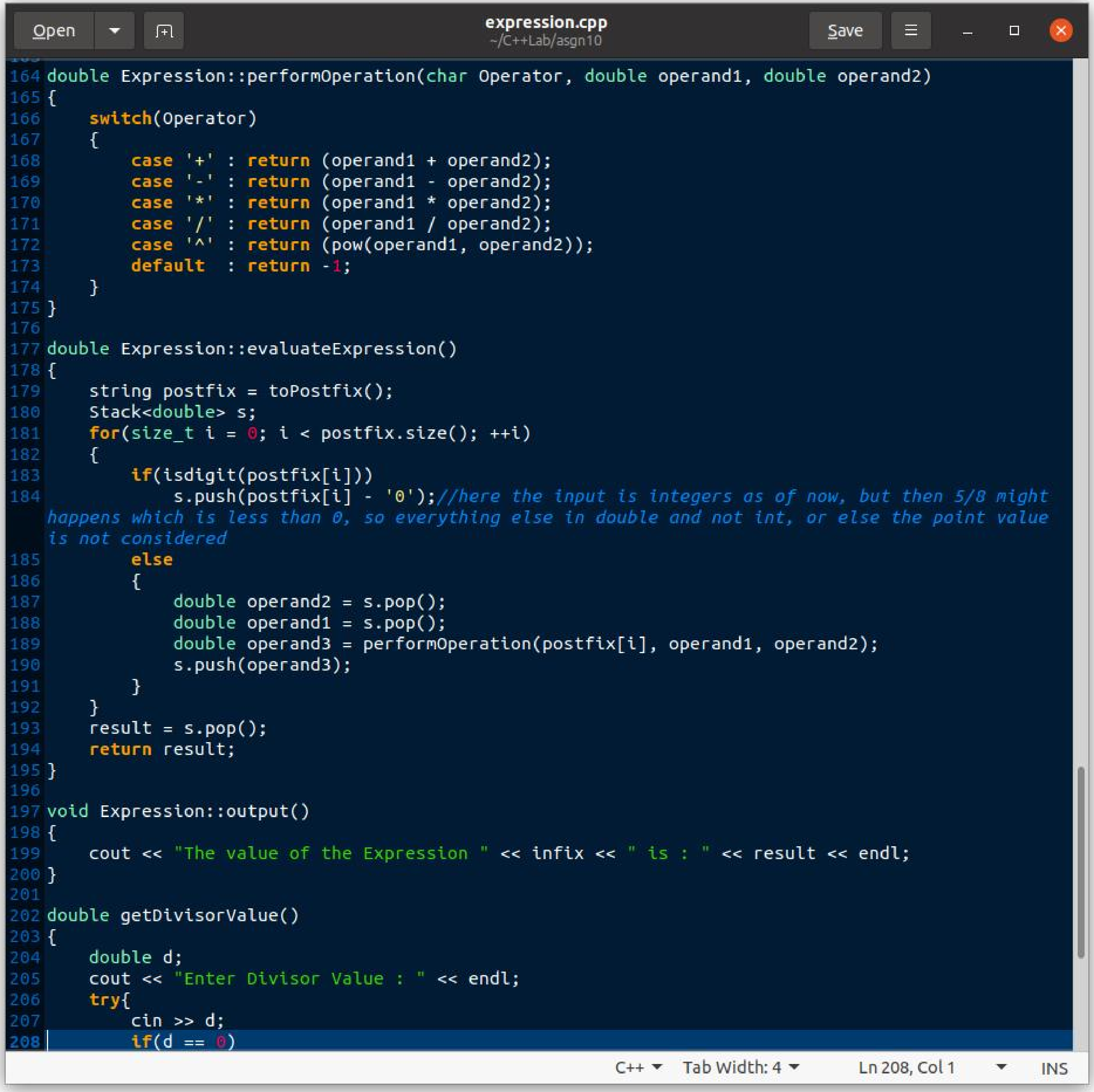
```cpp
class Expression
{
    private:
        string infix;
        double result;
        int isOperator(char);
        int isRightAssociative(char);
        int getPrecedenceValue(char);
        int hasGreaterPrecedence(char, char);
        int isOpeningBracket(char);
        int isClosingBracket(char);
        double performOperation(char, double, double);
    public:
        void getData();
        string toPostfix();
        double evaluateExpression();
        int getResult();
        void output();
};

void Expression::getData()
{
    cout << "Enter an Expression : ";
    cin >> infix;
}

int Expression::isOperator(char ch)
{
    switch(ch)
    {
        case '+' : return 1;
        case '-' : return 1;
        case '*' : return 1;
        case '/' : return 1;
        case '^' : return 1;
        default  : return 0;
    }
}

int Expression::isRightAssociative(char Operator)
{
    return (Operator == '^');
}

int Expression::getPrecedenceValue(char Operator)
{
    switch(Operator)
```



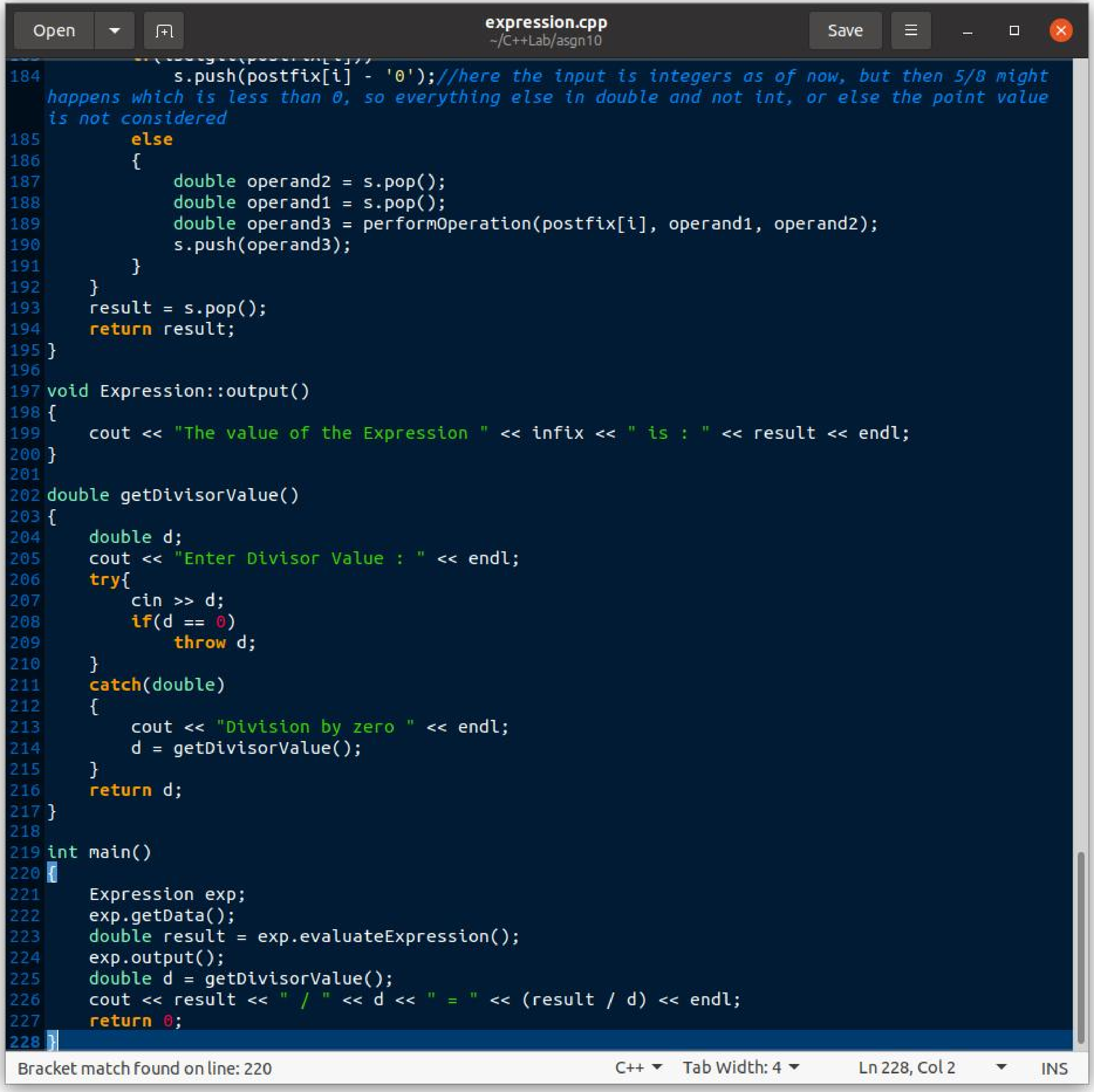
```cpp
int Expression::getPrecedenceValue(char Operator)
{
    switch(Operator)
    {
        case '+' : return 1;
        case '-' : return 1;
        case '*' : return 2;
        case '/' : return 2;
        case '^' : return 3;
        default  : return 0;
    }
}

int Expression::hasGreaterPrecedence(char Operator1, char Operator2)
{
    int precedenceValue1 = getPrecedenceValue(Operator1);
    int precedenceValue2 = getPrecedenceValue(Operator2);
    if(precedenceValue1 == precedenceValue2)
    {
        if(isRightAssociative(Operator1))
            return 0;
        return 1;
    }
    return (precedenceValue1 > precedenceValue2);
}

int Expression::isOpeningBracket(char ch)// can make these functions inline, by using if and || condition and operator
{
    switch(ch)
    {
        case '(' : return 1;
        case '{' : return 1;
        case '[' : return 1;
        default  : return 0;
    }
}

int Expression::isClosingBracket(char ch)
{
    switch(ch)
    {
        case ')' : return 1;
        case '}' : return 1;
        case ']' : return 1;
        default  : return 0;
    }
```



```cpp
int Expression::isClosingBracket(char ch)
{
    switch(ch)
    {
        case ')' : return 1;
        case '}' : return 1;
        case ']' : return 1;
        default  : return 0;
    }
}

string Expression::toPostfix()
{
    Stack<char> s;
    string postfix;
    for(size_t i = 0; i < infix.size(); ++i)
    {
        if(isdigit(infix[i]))
            postfix.push_back(infix[i]);
        else if(isOperator(infix[i]))
        {
            while(!s.isEmpty() && !isOpeningBracket(infix[i]) && hasGreaterPrecedence(s.Top(), infix[i]))
                postfix.push_back(s.pop());
            s.push(infix[i]);
        }
        else if(isOpeningBracket(infix[i]))
            s.push(infix[i]);
        else if(isClosingBracket(infix[i]))
        {
            while(!s.isEmpty() && !isOpeningBracket(s.Top()))
                postfix.push_back(s.pop());
            s.pop();
        }
    }
    while(!s.isEmpty())
        postfix.push_back(s.pop());
    return postfix;
}

double Expression::performOperation(char Operator, double operand1, double operand2)
{
    switch(Operator)
    {
        case '+' : return (operand1 + operand2);
        case '-' : return (operand1 - operand2);
        case '*' : return (operand1 * operand2);
```

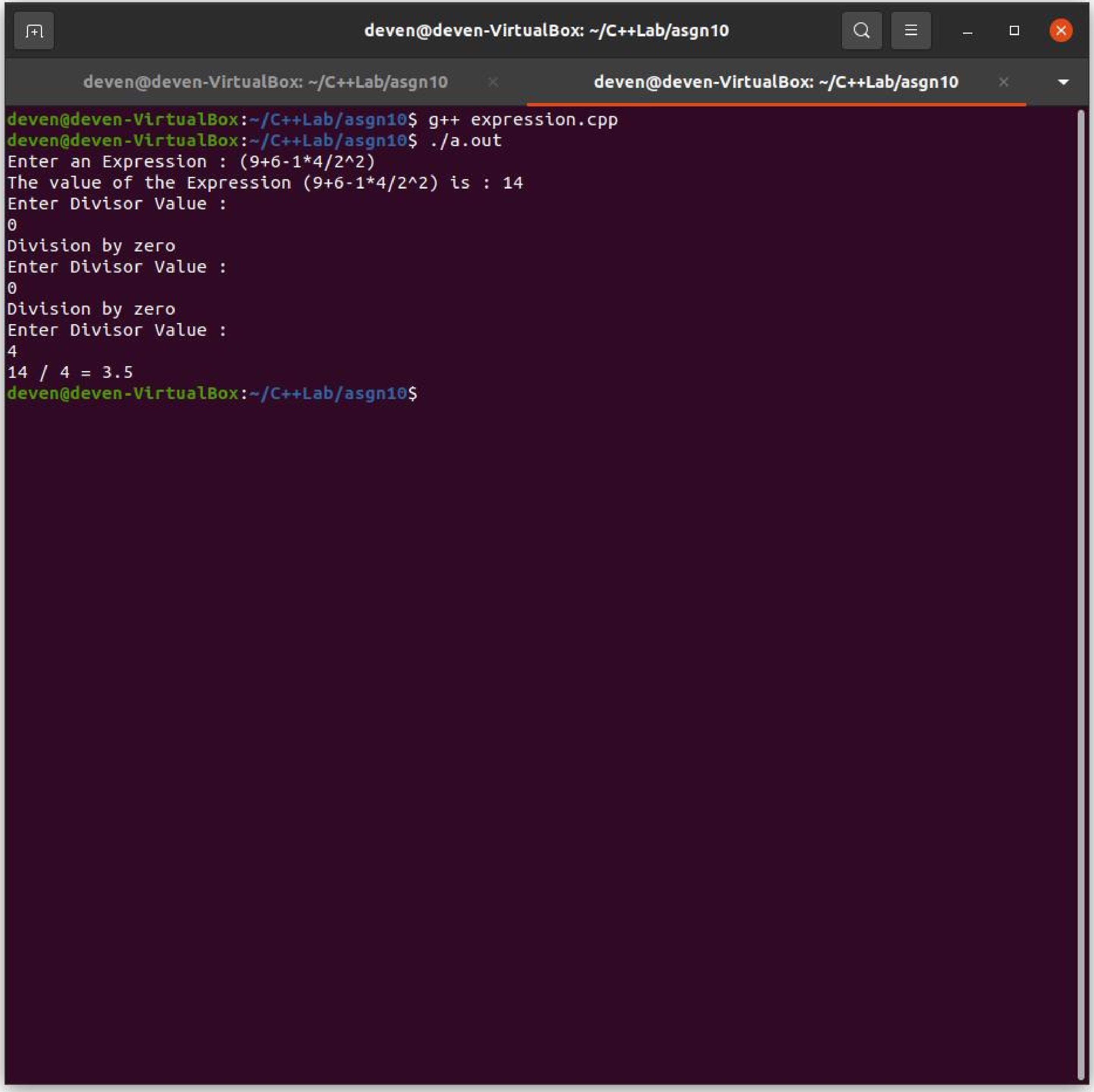


```cpp
double Expression::performOperation(char Operator, double operand1, double operand2)
{
    switch(Operator)
    {
        case '+' : return (operand1 + operand2);
        case '-' : return (operand1 - operand2);
        case '*' : return (operand1 * operand2);
        case '/' : return (operand1 / operand2);
        case '^' : return (pow(operand1, operand2));
        default  : return -1;
    }
}

double Expression::evaluateExpression()
{
    string postfix = toPostfix();
    Stack<double> s;
    for(size_t i = 0; i < postfix.size(); ++i)
    {
        if(isdigit(postfix[i]))
            s.push(postfix[i] - '0');//here the input is integers as of now, but then 5/8 might happens which is less than 0, so everything else in double and not int, or else the point value is not considered
        else
        {
            double operand2 = s.pop();
            double operand1 = s.pop();
            double operand3 = performOperation(postfix[i], operand1, operand2);
            s.push(operand3);
        }
    }
    result = s.pop();
    return result;
}

void Expression::output()
{
    cout << "The value of the Expression " << infix << " is : " << result << endl;
}

double getDivisorValue()
{
    double d;
    cout << "Enter Divisor Value : " << endl;
    try{
        cin >> d;
        if(d == 0)
```



```cpp
            s.push(postfix[i] - '0');//here the input is integers as of now, but then 5/8 might happens which is less than 0, so everything else in double and not int, or else the point value is not considered
        else
        {
            double operand2 = s.pop();
            double operand1 = s.pop();
            double operand3 = performOperation(postfix[i], operand1, operand2);
            s.push(operand3);
        }
    }
    result = s.pop();
    return result;
}

void Expression::output()
{
    cout << "The value of the Expression " << infix << " is : " << result << endl;
}

double getDivisorValue()
{
    double d;
    cout << "Enter Divisor Value : " << endl;
    try{
        cin >> d;
        if(d == 0)
            throw d;
    }
    catch(double)
    {
        cout << "Division by zero " << endl;
        d = getDivisorValue();
    }
    return d;
}

int main()
{
    Expression exp;
    exp.getData();
    double result = exp.evaluateExpression();
    exp.output();
    double d = getDivisorValue();
    cout << result << " / " << d << " = " << (result / d) << endl;
    return 0;
}
```

Bracket match found on line: 220



deven@deven-VirtualBox: ~/C++Lab/asgn10
deven@deven-VirtualBox: ~/C++Lab/asgn10
deven@deven-VirtualBox: ~/C++Lab/asgn10

```
deven@deven-VirtualBox:~/C++Lab/asgn10$ g++ expression.cpp
deven@deven-VirtualBox:~/C++Lab/asgn10$ ./a.out
Enter an Expression : (9+6-1*4/2^2)
The value of the Expression (9+6-1*4/2^2) is : 14
Enter Divisor Value :
0
Division by zero
Enter Divisor Value :
0
Division by zero
Enter Divisor Value :
4
14 / 4 = 3.5
deven@deven-VirtualBox:~/C++Lab/asgn10$
```