



SAVEETHA SCHOOL OF ENGINEERING
SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES



Engineer to Excel

SUB CODE & NAME: DSA01/ Object Oriented Programming with C++

LAB DAY 1/13-03-2024

EASY

1. Write a C++ program to declare and initialize a constant integer with value 10.

The screenshot shows a web browser window with the URL `programiz.com/cpp-programming/online-compiler/`. The page title is "Programiz C++ Online Compiler". A "C++ Certification" button is visible in the top right. The main area contains a code editor with the following C++ code:

```
1 #include <iostream>
2
3 int main() {
4     const int myConst = 10;
5     std::cout << "The value of the constant integer is: " << myConst << std::endl;
6     return 0;
7 }
8
```

To the right of the code editor is an "Output" window showing the result of the program execution:

```
/tmp/025Q2Xlmui.o
The value of the constant integer is: 10
```

2. Write a C++ program to demonstrate the use of integer, float, and character data types

The screenshot shows the Programiz C++ Online Compiler interface. The code editor contains the following C++ code:

```
1 #include <iostream>
2 int main()
3 {
4     int integerVar = 42;
5     float floatVar = 3.14;
6     char charVar = 'A';
7     std::cout << "Integer Variable: " << integerVar << std::endl;
8     std::cout << "Float Variable: " << floatVar << std::endl;
9     std::cout << "Character Variable: " << charVar << std::endl;
10    return 0;
11 }
12
```

The output window displays the following results:

```
/tmp/9J5A2bwes8.o
Integer Variable: 42
Float Variable: 3.14
Character Variable: A
```

3. Create a C++ program to declare an integer variable and a float variable, then assign the integer variable to the float variable

The screenshot shows the Programiz C++ Online Compiler interface. The code editor contains the following C++ code:

```
1 #include <iostream>
2 int main()
3 {
4     int intVar = 5;
5     float floatVar;
6     floatVar = intVar;
7     std::cout << "Integer Variable: " << intVar << std::endl;
8     std::cout << "Float Variable: " << floatVar << std::endl;
9     return 0;
10 }
11
```

The output window displays the following results:

```
/tmp/hZJa8ClkNT.o
Integer Variable: 5
Float Variable: 5
```

4. Implement a C++ program that performs an implicit conversion from integer to float

The screenshot shows the Programiz C++ Online Compiler interface. The code editor contains the following C++ code:

```
1 #include <iostream>
2 int main()
3 {
4     int intVar = 10;
5     float floatVar = intVar;
6     std::cout << "Integer Variable (intVar): " << intVar << std::endl;
7     std::cout << "Float Variable (floatVar) after implicit conversion: " <<
        floatVar << std::endl;
8     return 0;
9 }
10
```

The output window displays the following results:

```
/tmp/Lab38kLjNY.o
Integer Variable (intVar): 10
Float Variable (floatVar) after implicit conversion: 10
```

5. Write a C++ program to declare and initialize two integer variables and perform addition using the + operator

The screenshot shows the Programiz C++ Online Compiler interface. The code editor contains the following C++ code:

```
1 #include <iostream>
2 int main()
3 {
4     int num1 = 8;
5     int num2 = 15;
6     int sum = num1 + num2;
7     std::cout << "The sum of " << num1 << " and " << num2 << " is: " << sum <<
        std::endl;
8     return 0;
9 }
10
```

The output window displays the following result:

```
/tmp/IW9G9YjWtu.o
The sum of 8 and 15 is: 23
```

6. Create a C++ program to demonstrate the use of if-else control structure to determine if a number is even or odd

The screenshot shows the Programiz C++ Online Compiler interface. The code in main.cpp is as follows:

```
1 #include <iostream>
2 int main()
3 {
4     int a;
5     std::cout<<"Enter a number : ";
6     std::cin>>a;
7     if(a%2==0)
8     {
9         std::cout<<"Even";
10    }
11    else
12    {
13        std::cout<<"Odd";
14    }
15 }
16 }
17
```

The output on the right shows the execution result for input 34:

```
/tmp/Jr0KnIH4UC.o
Enter a number : 34
Even
```

7. Write a C++ program to declare an array of integers and display its elements using a loop.

The screenshot shows the Programiz C++ Online Compiler interface. The code in main.cpp is as follows:

```
1 #include <iostream>
2 int main()
3 {
4     int intArray[] = {1, 2, 3, 4, 5};
5     int arraySize = sizeof(intArray) / sizeof(intArray[0]);
6     std::cout << "Array elements are: ";
7     for (int i = 0; i < arraySize; i++) {
8         std::cout << intArray[i] << " ";
9     }
10    std::cout << std::endl;
11    return 0;
12 }
13
```

The output on the right shows the execution result:

```
/tmp/Qct4HvsVlF.o
Array elements are: 1 2 3 4 5
```

8. Implement a C++ program to demonstrate the use of switch-case control structure to display the name of the day based on the user input (1-7).

The screenshot shows a web browser with the URL `programiz.com/cpp-programming/online-compiler/`. The page title is "Programiz C++ Online Compiler". On the right, there is a "C++ Certification" button. The main area contains a code editor with a file named `main.cpp`. The code is as follows:

```
1 #include <iostream>
2 int main()
3 {
4     int dayNumber;
5     std::cout << "Enter a number (1-7) to get the corresponding day of the week
6     : ";
7     std::cin >> dayNumber;
8     switch(dayNumber) {
9         case 1:
10         std::cout << "The day is Sunday." << std::endl;
11         break;
12         case 2:
13         std::cout << "The day is Monday." << std::endl;
14         break;
15         case 3:
16         std::cout << "The day is Tuesday." << std::endl;
17         break;
18         case 4:
19         std::cout << "The day is Wednesday." << std::endl;
20         break;
21         case 5:
22         std::cout << "The day is Thursday." << std::endl;
23         break;
24         case 6:
25         std::cout << "The day is Friday." << std::endl;
26         break;
27     }
```

The "Output" panel on the right shows the execution results:

```
/tmp/BZUMNPg9DN.o
Enter a number (1-7) to get the corresponding day of the week: 5
The day is Thursday.
```

9. Create a C++ program to calculate the area of a rectangle using user input for length and width

The screenshot shows the same online C++ compiler interface. The code in `main.cpp` is as follows:

```
1 #include <iostream>
2 int main()
3 {
4     double length, width, area;
5     std::cout << "Enter the length of the rectangle: ";
6     std::cin >> length;
7     std::cout << "Enter the width of the rectangle: ";
8     std::cin >> width;
9     area = length * width;
10    std::cout << "The area of the rectangle is: " << area << std::endl;
11    return 0;
12 }
```

The "Output" panel shows the execution results:

```
/tmp/gCK900eIL9.o
Enter the length of the rectangle: 5
Enter the width of the rectangle: 10
The area of the rectangle is: 50
```

10. Write a C++ program to find the maximum of three numbers using conditional statements.

The screenshot shows the Programiz C++ Online Compiler interface. The code in `main.cpp` is as follows:

```
1 #include <iostream>
2 int main()
3 {
4     int num1, num2, num3, max;
5     std::cout << "Enter three numbers: ";
6     std::cin >> num1 >> num2 >> num3;
7     if (num1 >= num2 && num1 >= num3)
8     {
9         max = num1;
10    } else if (num2 >= num1 && num2 >= num3)
11    {
12        max = num2;
13    } else
14    {
15        max = num3;
16    }
17    std::cout << "The maximum number is: " << max << std::endl;
18    return 0;
19 }
20
```

The output of the program is:

```
/tmp/Q096J6paJZ.o
Enter three numbers: 3
6
1
The maximum number is: 6
```

MEDIUM

1. Write a C++ program that calculates the area of a circle. Use a constant variable for the value of pi and prompt the user to enter the radius. Display the calculated area.

The screenshot shows the Programiz C++ Online Compiler interface. The code in `main.cpp` is as follows:

```
1 #include <iostream>
2 int main()
3 {
4     int r,a;
5     std::cout<<"Enter the radius of the circle : ";
6     std::cin>>r;
7     a=3.14*(r*r);
8     std::cout<<"Area of the circle is :";
9     std::cout<<a;
10 }
11
```

The output of the program is:

```
/tmp/7zjMd9Jgfp.o
Enter the radius of the circle : 7
Area of the circle is :153
```

2. Develop a C++ program that reads two numbers from the user, one integer and one float. Perform division of the float by the integer and display the result. Ensure proper type compatibility and implicit conversion handling.

The screenshot shows the Programiz C++ Online Compiler interface. The code in `main.cpp` is as follows:

```
1 #include <iostream>
2 int main()
3 {
4     int integerInput;
5     float floatInput, result;
6     std::cout << "Enter an integer: ";
7     std::cin >> integerInput;
8     std::cout << "Enter a float: ";
9     std::cin >> floatInput;
10    result = floatInput / static_cast<float>(integerInput);
11    std::cout << "The result of the division is: " << result << std::endl;
12    return 0;
13 }
```

The output of the program is:

```
/tmp/ly2XnD2XJ6.o
Enter an integer: 5
Enter a float: 7.90
The result of the division is: 1.58
```

3. Write a C++ program to determine whether a given year is a leap year or not. Use logical operators and appropriate control structures to implement the leap year logic.

The screenshot shows the Programiz C++ Online Compiler interface. The code in `main.cpp` is as follows:

```
1 #include <iostream>
2 int main()
3 {
4     int year;
5     std::cout << "Enter a year: ";
6     std::cin >> year;
7     if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0))
8     {
9         std::cout << year << " is a leap year." << std::endl;
10    } else
11    {
12        std::cout << year << " is not a leap year." << std::endl;
13    }
14    return 0;
15 }
```

The output of the program is:

```
/tmp/T9Zt7cMKop.o
Enter a year: 2024
2024 is a leap year.
```

4. Create a C++ program that calculates the area of a rectangle. Prompt the user to enter the length and width of the rectangle as floating-point numbers, and display the calculated area.

The screenshot shows the Programiz C++ Online Compiler interface. The code editor contains a C++ program that prompts the user for the length and width of a rectangle and calculates its area. The output window shows the program's execution with the user inputting 20 for length and 10 for width, resulting in an area of 200.

```
main.cpp
1 #include <iostream>
2 int main()
3 {
4     float length, width, area;
5     std::cout << "Please enter the length of the rectangle: ";
6     std::cin >> length;
7     std::cout << "Please enter the width of the rectangle: ";
8     std::cin >> width;
9     area = length * width;
10    std::cout << "The area of the rectangle is: " << area << std::endl;
11    return 0;
12 }
13
```

Output

```
/tmp/t3y4BUMbwW.o
Please enter the length of the rectangle: 20
Please enter the width of the rectangle: 10
The area of the rectangle is: 200
```

5. Develop a C++ program that reads an integer from the user and checks if it is an odd number. Use bitwise AND operator and handle implicit conversion properly.

The screenshot shows the Programiz C++ Online Compiler interface. The code editor contains a C++ program that prompts the user for an integer and checks if it is odd or even using the bitwise AND operator. The output window shows the program's execution with the user inputting 5, resulting in the message "The number is odd."

```
main.cpp
1 #include <iostream>
2 int main()
3 {
4     int number;
5     std::cout << "Enter an integer: ";
6     std::cin >> number;
7     if (number & 1)
8     {
9         std::cout << "The number is odd." << std::endl;
10    } else
11    {
12        std::cout << "The number is even." << std::endl;
13    }
14    return 0;
15 }
16
```

Output

```
/tmp/R0NUgzzqIAj.o
Enter an integer: 5
The number is odd.
```

6. Write a C++ program that prompts the user to enter a month number (1-12) and displays the corresponding month name. Use a switch-case statement to implement this.

The screenshot shows the Programiz C++ Online Compiler interface. The code in `main.cpp` is as follows:

```
1 #include <iostream>
2 int main()
3 {
4     int monthNumber;
5     std::cout << "Enter a month number (1-12): ";
6     std::cin >> monthNumber;
7     switch (monthNumber) {
8         case 1:
9             std::cout << "January" << std::endl;
10            break;
11        case 2:
12            std::cout << "February" << std::endl;
13            break;
14        case 3:
15            std::cout << "March" << std::endl;
16            break;
17        case 4:
18            std::cout << "April" << std::endl;
19            break;
20        case 5:
21            std::cout << "May" << std::endl;
22            break;
23        case 6:
24            std::cout << "June" << std::endl;
25            break;
26        case 7:
```

The output window shows the result of running the program with input 8:

```
/tmp/M0rDZHONtQ.o
Enter a month number (1-12): 8
August
```

7. Create a C++ program that calculates the volume of a sphere. Prompt the user to enter the radius and use a constant variable for the value of pi. Display the calculated volume.

The screenshot shows the Programiz C++ Online Compiler interface. The code in `main.cpp` is as follows:

```
1 #include <iostream>
2 #include <cmath>
3 int main()
4 {
5     const double PI = M_PI;
6     double radius, volume;
7     std::cout << "Please enter the radius of the sphere: ";
8     std::cin >> radius;
9     volume = (4.0 / 3.0) * PI * pow(radius, 3);
10    std::cout << "The volume of the sphere is: " << volume << std::endl;
11    return 0;
12 }
13
```

The output window shows the result of running the program with input 7:

```
/tmp/13y9TU2816.o
Please enter the radius of the sphere: 7
The volume of the sphere is: 1436.76
```

8. Develop a C++ program that reads two integers from the user and performs division. Ensure that if the second number is zero, the program should not crash, but display an appropriate error message.

The screenshot shows the Programiz Online C++ Compiler interface. The code in `main.cpp` is as follows:

```

1 #include <iostream>
2 int main()
3 {
4     int numerator, denominator;
5     double result;
6     std::cout << "Enter the numerator: ";
7     std::cin >> numerator;
8     std::cout << "Enter the denominator: ";
9     std::cin >> denominator;
10    if (denominator != 0)
11    {
12        result = static_cast<double>(numerator) / denominator;
13        std::cout << "The result of the division is: " << result << std::endl;
14    } else
15    {
16        std::cout << "Error: Division by zero is not allowed." << std::endl;
17    }
18    return 0;
19 }
20

```

The output of the program is:

```

/tmp/xU2nJ4VnGV.o
Enter the numerator: 25
Enter the denominator: 5
The result of the division is: 5

```

- Write a C++ program that defines a class representing a complex number. Implement operator overloading for addition and subtraction of complex numbers.

The screenshot shows the Programiz Online C++ Compiler interface. The code in `main.cpp` is as follows:

```

1 #include <iostream>
2 class Complex
3 {
4 public:
5     double real;
6     double imag;
7     Complex(double r = 0.0, double i = 0.0) : real(r), imag(i) {}
8     Complex operator+(const Complex& rhs) const {
9         return Complex(real + rhs.real, imag + rhs.imag);
10    }
11    Complex operator-(const Complex& rhs) const {
12        return Complex(real - rhs.real, imag - rhs.imag);
13    }
14    void display() const {
15        std::cout << real << " + " << imag << "i" << std::endl;
16    }
17 };
18 int main() {
19     Complex c1(5.0, 4.0), c2(2.0, 3.5);
20     Complex c3, c4;
21     c3 = c1 + c2;
22     std::cout << "Addition of the complex numbers: ";
23     c3.display();
24     c4 = c1 - c2;
25     std::cout << "Subtraction of the complex numbers: ";
26     c4.display();
27 }

```

The output of the program is:

```

/tmp/A3sYH87sEc.o
Addition of the complex numbers: 7 + 7.5i
Subtraction of the complex numbers: 3 + 0.5i

```

- Create a C++ program that prompts the user to enter a number and calculates the factorial of that number using a loop. Display the result.

The screenshot shows the Programiz C++ Online Compiler interface. The code in main.cpp is as follows:

```
1 #include <iostream>
2 int main()
3 {
4     unsigned int number;
5     unsigned long long factorial = 1;
6     std::cout << "Enter a positive integer: ";
7     std::cin >> number;
8     for(unsigned int i = 1; i <= number; ++i) {
9         factorial *= i;
10    }
11    std::cout << "Factorial of " << number << " is: " << factorial << std::endl;
12    return 0;
13 }
```

The output on the right shows the program's execution:

```
/tmp/vRZXn8DyWl.o
Enter a positive integer: 5
Factorial of 5 is: 120
```

HARD

1. Write a C++ program to implement a recursive function to calculate the Fibonacci sequence up to a given number 'n'.

The screenshot shows the Programiz C++ Online Compiler interface. The code in main.cpp is as follows:

```
1 #include <iostream>
2 int fibonacci(int n)
3 {
4     if (n <= 1)
5     {
6         return n;
7     } else
8     {
9         return fibonacci(n - 1) + fibonacci(n - 2);
10    }
11 }
12 int main() {
13     int n;
14     std::cout << "Enter the number of terms for Fibonacci sequence: ";
15     std::cin >> n;
16     std::cout << "Fibonacci sequence up to " << n << " terms:" << std::endl;
17     for (int i = 0; i < n; i++) {
18         std::cout << fibonacci(i) << " ";
19     }
20     std::cout << std::endl;
21     return 0;
22 }
```

The output on the right shows the program's execution:

```
/tmp/mhAew2VqKw.o
Enter the number of terms for Fibonacci sequence: 7
Fibonacci sequence up to 7 terms:
0 1 1 2 3 5 8
```

2. Develop a C++ program that dynamically allocates memory for an array of integers based on user input for the array size. Implement error handling for memory allocation failure.

The screenshot shows a web browser with the URL `programiz.com/cpp-programming/online-compiler/`. The page title is "Programiz C++ Online Compiler". The code editor contains the following C++ code:

```

6 int size;
7 std::cout << "Enter the size of the array: ";
8 std::cin >> size;
9 try
10 {
11     array = new int[size];
12     std::cout << "Enter " << size << " integers:" << std::endl;
13     for (int i = 0; i < size; ++i)
14     {
15         std::cin >> array[i];
16     }
17     std::cout << "Array elements are:" << std::endl;
18     for (int i = 0; i < size; ++i)
19     {
20         std::cout << array[i] << " ";
21     }
22     std::cout << std::endl;
23     delete[] array;
24 }
25 catch (const std::bad_alloc& e)
26 {
27     std::cerr << "Memory allocation failed: " << e.what() << std::endl;
28 }
29 return 0;
30 }
31

```

The output window shows the following text:

```

/tmp/SCnvB8sEhA.o
Enter the size of the array: 5
Enter 5 integers:
3
7
8
3
5
Array elements are:
3 7 8 3 5

```

3. Create a C++ program that defines a class representing a matrix. Implement operator overloading for matrix addition, subtraction, and multiplication.

The screenshot shows the same online C++ compiler with the following C++ code:

```

1 #include <iostream>
2 #include <vector>
3 class Matrix {
4 private:
5     std::vector<std::vector<int>> mat;
6     unsigned rows, cols;
7 public:
8     Matrix(unsigned r, unsigned c) : rows(r), cols(c), mat(r, std::vector<int>
9         >(c, 0)) {}
10     Matrix operator+(const Matrix& rhs) const {
11         if (rows != rhs.rows || cols != rhs.cols) {
12             throw std::invalid_argument("Matrices dimensions mismatch for
13                 addition");
14         }
15         Matrix result(rows, cols);
16         for (unsigned i = 0; i < rows; ++i) {
17             for (unsigned j = 0; j < cols; ++j) {
18                 result.mat[i][j] = mat[i][j] + rhs.mat[i][j];
19             }
20         }
21         return result;
22     }
23     Matrix operator-(const Matrix& rhs) const {
24         if (rows != rhs.rows || cols != rhs.cols) {
25             throw std::invalid_argument("Matrices dimensions mismatch for
26                 subtraction");
27         }
28         Matrix result(rows, cols);
29         for (unsigned i = 0; i < rows; ++i) {
30             for (unsigned j = 0; j < cols; ++j) {
31                 result.mat[i][j] = mat[i][j] - rhs.mat[i][j];
32             }
33         }
34         return result;
35     }
36     Matrix operator*(const Matrix& rhs) const {
37         if (cols != rhs.rows) {
38             throw std::invalid_argument("Matrices dimensions mismatch for
39                 multiplication");
40         }
41         Matrix result(rows, rhs.cols);
42         for (unsigned i = 0; i < rows; ++i) {
43             for (unsigned j = 0; j < rhs.cols; ++j) {
44                 int sum = 0;
45                 for (unsigned k = 0; k < cols; ++k) {
46                     sum += mat[i][k] * rhs.mat[k][j];
47                 }
48                 result.mat[i][j] = sum;
49             }
50         }
51         return result;
52     }
53 }
54 int main() {
55     Matrix m1(2, 2);
56     Matrix m2(2, 2);
57     Matrix m3(2, 2);
58     Matrix m4(2, 2);
59     Matrix m5(2, 2);
60     Matrix m6(2, 2);
61     Matrix m7(2, 2);
62     Matrix m8(2, 2);
63     Matrix m9(2, 2);
64     Matrix m10(2, 2);
65     Matrix m11(2, 2);
66     Matrix m12(2, 2);
67     Matrix m13(2, 2);
68     Matrix m14(2, 2);
69     Matrix m15(2, 2);
70     Matrix m16(2, 2);
71     Matrix m17(2, 2);
72     Matrix m18(2, 2);
73     Matrix m19(2, 2);
74     Matrix m20(2, 2);
75     Matrix m21(2, 2);
76     Matrix m22(2, 2);
77     Matrix m23(2, 2);
78     Matrix m24(2, 2);
79     Matrix m25(2, 2);
80     Matrix m26(2, 2);
81     Matrix m27(2, 2);
82     Matrix m28(2, 2);
83     Matrix m29(2, 2);
84     Matrix m30(2, 2);
85     Matrix m31(2, 2);
86     Matrix m32(2, 2);
87     Matrix m33(2, 2);
88     Matrix m34(2, 2);
89     Matrix m35(2, 2);
90     Matrix m36(2, 2);
91     Matrix m37(2, 2);
92     Matrix m38(2, 2);
93     Matrix m39(2, 2);
94     Matrix m40(2, 2);
95     Matrix m41(2, 2);
96     Matrix m42(2, 2);
97     Matrix m43(2, 2);
98     Matrix m44(2, 2);
99     Matrix m45(2, 2);
100    Matrix m46(2, 2);
101    Matrix m47(2, 2);
102    Matrix m48(2, 2);
103    Matrix m49(2, 2);
104    Matrix m50(2, 2);
105    Matrix m51(2, 2);
106    Matrix m52(2, 2);
107    Matrix m53(2, 2);
108    Matrix m54(2, 2);
109    Matrix m55(2, 2);
110    Matrix m56(2, 2);
111    Matrix m57(2, 2);
112    Matrix m58(2, 2);
113    Matrix m59(2, 2);
114    Matrix m60(2, 2);
115    Matrix m61(2, 2);
116    Matrix m62(2, 2);
117    Matrix m63(2, 2);
118    Matrix m64(2, 2);
119    Matrix m65(2, 2);
120    Matrix m66(2, 2);
121    Matrix m67(2, 2);
122    Matrix m68(2, 2);
123    Matrix m69(2, 2);
124    Matrix m70(2, 2);
125    Matrix m71(2, 2);
126    Matrix m72(2, 2);
127    Matrix m73(2, 2);
128    Matrix m74(2, 2);
129    Matrix m75(2, 2);
130    Matrix m76(2, 2);
131    Matrix m77(2, 2);
132    Matrix m78(2, 2);
133    Matrix m79(2, 2);
134    Matrix m80(2, 2);
135    Matrix m81(2, 2);
136    Matrix m82(2, 2);
137    Matrix m83(2, 2);
138    Matrix m84(2, 2);
139    Matrix m85(2, 2);
140    Matrix m86(2, 2);
141    Matrix m87(2, 2);
142    Matrix m88(2, 2);
143    Matrix m89(2, 2);
144    Matrix m90(2, 2);
145    Matrix m91(2, 2);
146    Matrix m92(2, 2);
147    Matrix m93(2, 2);
148    Matrix m94(2, 2);
149    Matrix m95(2, 2);
150    Matrix m96(2, 2);
151    Matrix m97(2, 2);
152    Matrix m98(2, 2);
153    Matrix m99(2, 2);
154    Matrix m100(2, 2);
155    Matrix m101(2, 2);
156    Matrix m102(2, 2);
157    Matrix m103(2, 2);
158    Matrix m104(2, 2);
159    Matrix m105(2, 2);
160    Matrix m106(2, 2);
161    Matrix m107(2, 2);
162    Matrix m108(2, 2);
163    Matrix m109(2, 2);
164    Matrix m110(2, 2);
165    Matrix m111(2, 2);
166    Matrix m112(2, 2);
167    Matrix m113(2, 2);
168    Matrix m114(2, 2);
169    Matrix m115(2, 2);
170    Matrix m116(2, 2);
171    Matrix m117(2, 2);
172    Matrix m118(2, 2);
173    Matrix m119(2, 2);
174    Matrix m120(2, 2);
175    Matrix m121(2, 2);
176    Matrix m122(2, 2);
177    Matrix m123(2, 2);
178    Matrix m124(2, 2);
179    Matrix m125(2, 2);
180    Matrix m126(2, 2);
181    Matrix m127(2, 2);
182    Matrix m128(2, 2);
183    Matrix m129(2, 2);
184    Matrix m130(2, 2);
185    Matrix m131(2, 2);
186    Matrix m132(2, 2);
187    Matrix m133(2, 2);
188    Matrix m134(2, 2);
189    Matrix m135(2, 2);
190    Matrix m136(2, 2);
191    Matrix m137(2, 2);
192    Matrix m138(2, 2);
193    Matrix m139(2, 2);
194    Matrix m140(2, 2);
195    Matrix m141(2, 2);
196    Matrix m142(2, 2);
197    Matrix m143(2, 2);
198    Matrix m144(2, 2);
199    Matrix m145(2, 2);
200    Matrix m146(2, 2);
201    Matrix m147(2, 2);
202    Matrix m148(2, 2);
203    Matrix m149(2, 2);
204    Matrix m150(2, 2);
205    Matrix m151(2, 2);
206    Matrix m152(2, 2);
207    Matrix m153(2, 2);
208    Matrix m154(2, 2);
209    Matrix m155(2, 2);
210    Matrix m156(2, 2);
211    Matrix m157(2, 2);
212    Matrix m158(2, 2);
213    Matrix m159(2, 2);
214    Matrix m160(2, 2);
215    Matrix m161(2, 2);
216    Matrix m162(2, 2);
217    Matrix m163(2, 2);
218    Matrix m164(2, 2);
219    Matrix m165(2, 2);
220    Matrix m166(2, 2);
221    Matrix m167(2, 2);
222    Matrix m168(2, 2);
223    Matrix m169(2, 2);
224    Matrix m170(2, 2);
225    Matrix m171(2, 2);
226    Matrix m172(2, 2);
227    Matrix m173(2, 2);
228    Matrix m174(2, 2);
229    Matrix m175(2, 2);
230    Matrix m176(2, 2);
231    Matrix m177(2, 2);
232    Matrix m178(2, 2);
233    Matrix m179(2, 2);
234    Matrix m180(2, 2);
235    Matrix m181(2, 2);
236    Matrix m182(2, 2);
237    Matrix m183(2, 2);
238    Matrix m184(2, 2);
239    Matrix m185(2, 2);
240    Matrix m186(2, 2);
241    Matrix m187(2, 2);
242    Matrix m188(2, 2);
243    Matrix m189(2, 2);
244    Matrix m190(2, 2);
245    Matrix m191(2, 2);
246    Matrix m192(2, 2);
247    Matrix m193(2, 2);
248    Matrix m194(2, 2);
249    Matrix m195(2, 2);
250    Matrix m196(2, 2);
251    Matrix m197(2, 2);
252    Matrix m198(2, 2);
253    Matrix m199(2, 2);
254    Matrix m200(2, 2);
255    Matrix m201(2, 2);
256    Matrix m202(2, 2);
257    Matrix m203(2, 2);
258    Matrix m204(2, 2);
259    Matrix m205(2, 2);
260    Matrix m206(2, 2);
261    Matrix m207(2, 2);
262    Matrix m208(2, 2);
263    Matrix m209(2, 2);
264    Matrix m210(2, 2);
265    Matrix m211(2, 2);
266    Matrix m212(2, 2);
267    Matrix m213(2, 2);
268    Matrix m214(2, 2);
269    Matrix m215(2, 2);
270    Matrix m216(2, 2);
271    Matrix m217(2, 2);
272    Matrix m218(2, 2);
273    Matrix m219(2, 2);
274    Matrix m220(2, 2);
275    Matrix m221(2, 2);
276    Matrix m222(2, 2);
277    Matrix m223(2, 2);
278    Matrix m224(2, 2);
279    Matrix m225(2, 2);
280    Matrix m226(2, 2);
281    Matrix m227(2, 2);
282    Matrix m228(2, 2);
283    Matrix m229(2, 2);
284    Matrix m230(2, 2);
285    Matrix m231(2, 2);
286    Matrix m232(2, 2);
287    Matrix m233(2, 2);
288    Matrix m234(2, 2);
289    Matrix m235(2, 2);
290    Matrix m236(2, 2);
291    Matrix m237(2, 2);
292    Matrix m238(2, 2);
293    Matrix m239(2, 2);
294    Matrix m240(2, 2);
295    Matrix m241(2, 2);
296    Matrix m242(2, 2);
297    Matrix m243(2, 2);
298    Matrix m244(2, 2);
299    Matrix m245(2, 2);
300    Matrix m246(2, 2);
301    Matrix m247(2, 2);
302    Matrix m248(2, 2);
303    Matrix m249(2, 2);
304    Matrix m250(2, 2);
305    Matrix m251(2, 2);
306    Matrix m252(2, 2);
307    Matrix m253(2, 2);
308    Matrix m254(2, 2);
309    Matrix m255(2, 2);
310    Matrix m256(2, 2);
311    Matrix m257(2, 2);
312    Matrix m258(2, 2);
313    Matrix m259(2, 2);
314    Matrix m260(2, 2);
315    Matrix m261(2, 2);
316    Matrix m262(2, 2);
317    Matrix m263(2, 2);
318    Matrix m264(2, 2);
319    Matrix m265(2, 2);
320    Matrix m266(2, 2);
321    Matrix m267(2, 2);
322    Matrix m268(2, 2);
323    Matrix m269(2, 2);
324    Matrix m270(2, 2);
325    Matrix m271(2, 2);
326    Matrix m272(2, 2);
327    Matrix m273(2, 2);
328    Matrix m274(2, 2);
329    Matrix m275(2, 2);
330    Matrix m276(2, 2);
331    Matrix m277(2, 2);
332    Matrix m278(2, 2);
333    Matrix m279(2, 2);
334    Matrix m280(2, 2);
335    Matrix m281(2, 2);
336    Matrix m282(2, 2);
337    Matrix m283(2, 2);
338    Matrix m284(2, 2);
339    Matrix m285(2, 2);
340    Matrix m286(2, 2);
341    Matrix m287(2, 2);
342    Matrix m288(2, 2);
343    Matrix m289(2, 2);
344    Matrix m290(2, 2);
345    Matrix m291(2, 2);
346    Matrix m292(2, 2);
347    Matrix m293(2, 2);
348    Matrix m294(2, 2);
349    Matrix m295(2, 2);
350    Matrix m296(2, 2);
351    Matrix m297(2, 2);
352    Matrix m298(2, 2);
353    Matrix m299(2, 2);
354    Matrix m300(2, 2);
355    Matrix m301(2, 2);
356    Matrix m302(2, 2);
357    Matrix m303(2, 2);
358    Matrix m304(2, 2);
359    Matrix m305(2, 2);
360    Matrix m306(2, 2);
361    Matrix m307(2, 2);
362    Matrix m308(2, 2);
363    Matrix m309(2, 2);
364    Matrix m310(2, 2);
365    Matrix m311(2, 2);
366    Matrix m312(2, 2);
367    Matrix m313(2, 2);
368    Matrix m314(2, 2);
369    Matrix m315(2, 2);
370    Matrix m316(2, 2);
371    Matrix m317(2, 2);
372    Matrix m318(2, 2);
373    Matrix m319(2, 2);
374    Matrix m320(2, 2);
375    Matrix m321(2, 2);
376    Matrix m322(2, 2);
377    Matrix m323(2, 2);
378    Matrix m324(2, 2);
379    Matrix m325(2, 2);
380    Matrix m326(2, 2);
381    Matrix m327(2, 2);
382    Matrix m328(2, 2);
383    Matrix m329(2, 2);
384    Matrix m330(2, 2);
385    Matrix m331(2, 2);
386    Matrix m332(2, 2);
387    Matrix m333(2, 2);
388    Matrix m334(2, 2);
389    Matrix m335(2, 2);
390    Matrix m336(2, 2);
391    Matrix m337(2, 2);
392    Matrix m338(2, 2);
393    Matrix m339(2, 2);
394    Matrix m340(2, 2);
395    Matrix m341(2, 2);
396    Matrix m342(2, 2);
397    Matrix m343(2, 2);
398    Matrix m344(2, 2);
399    Matrix m345(2, 2);
400    Matrix m346(2, 2);
401    Matrix m347(2, 2);
402    Matrix m348(2, 2);
403    Matrix m349(2, 2);
404    Matrix m350(2, 2);
405    Matrix m351(2, 2);
406    Matrix m352(2, 2);
407    Matrix m353(2, 2);
408    Matrix m354(2, 2);
409    Matrix m355(2, 2);
410    Matrix m356(2, 2);
411    Matrix m357(2, 2);
412    Matrix m358(2, 2);
413    Matrix m359(2, 2);
414    Matrix m360(2, 2);
415    Matrix m361(2, 2);
416    Matrix m362(2, 2);
417    Matrix m363(2, 2);
418    Matrix m364(2, 2);
419    Matrix m365(2, 2);
420    Matrix m366(2, 2);
421    Matrix m367(2, 2);
422    Matrix m368(2, 2);
423    Matrix m369(2, 2);
424    Matrix m370(2, 2);
425    Matrix m371(2, 2);
426    Matrix m372(2, 2);
427    Matrix m373(2, 2);
428    Matrix m374(2, 2);
429    Matrix m375(2, 2);
430    Matrix m376(2, 2);
431    Matrix m377(2, 2);
432    Matrix m378(2, 2);
433    Matrix m379(2, 2);
434    Matrix m380(2, 2);
435    Matrix m381(2, 2);
436    Matrix m382(2, 2);
437    Matrix m383(2, 2);
438    Matrix m384(2, 2);
439    Matrix m385(2, 2);
440    Matrix m386(2, 2);
441    Matrix m387(2, 2);
442    Matrix m388(2, 2);
443    Matrix m389(2, 2);
444    Matrix m390(2, 2);
445    Matrix m391(2, 2);
446    Matrix m392(2, 2);
447    Matrix m393(2, 2);
448    Matrix m394(2, 2);
449    Matrix m395(2, 2);
450    Matrix m396(2, 2);
451    Matrix m397(2, 2);
452    Matrix m398(2, 2);
453    Matrix m399(2, 2);
454    Matrix m400(2, 2);
455    Matrix m401(2, 2);
456    Matrix m402(2, 2);
457    Matrix m403(2, 2);
458    Matrix m404(2, 2);
459    Matrix m405(2, 2);
460    Matrix m406(2, 2);
461    Matrix m407(2, 2);
462    Matrix m408(2, 2);
463    Matrix m409(2, 2);
464    Matrix m410(2, 2);
465    Matrix m411(2, 2);
466    Matrix m412(2, 2);
467    Matrix m413(2, 2);
468    Matrix m414(2, 2);
469    Matrix m415(2, 2);
470    Matrix m416(2, 2);
471    Matrix m417(2, 2);
472    Matrix m418(2, 2);
473    Matrix m419(2, 2);
474    Matrix m420(2, 2);
475    Matrix m421(2, 2);
476    Matrix m422(2, 2);
477    Matrix m423(2, 2);
478    Matrix m424(2, 2);
479    Matrix m425(2, 2);
480    Matrix m426(2, 2);
481    Matrix m427(2, 2);
482    Matrix m428(2, 2);
483    Matrix m429(2, 2);
484    Matrix m430(2, 2);
485    Matrix m431(2, 2);
486    Matrix m432(2, 2);
487    Matrix m433(2, 2);
488    Matrix m434(2, 2);
489    Matrix m435(2, 2);
490    Matrix m436(2, 2);
491    Matrix m437(2, 2);
492    Matrix m438(2, 2);
493    Matrix m439(2, 2);
494    Matrix m440(2, 2);
495    Matrix m441(2, 2);
496    Matrix m442(2, 2);
497    Matrix m443(2, 2);
498    Matrix m444(2, 2);
499    Matrix m445(2, 2);
500    Matrix m446(2, 2);
501    Matrix m447(2, 2);
502    Matrix m448(2, 2);
503    Matrix m449(2, 2);
504    Matrix m450(2, 2);
505    Matrix m451(2, 2);
506    Matrix m452(2, 2);
507    Matrix m453(2, 2);
508    Matrix m454(2, 2);
509    Matrix m455(2, 2);
510    Matrix m456(2, 2);
511    Matrix m457(2, 2);
512    Matrix m458(2, 2);
513    Matrix m459(2, 2);
514    Matrix m460(2, 2);
515    Matrix m461(2, 2);
516    Matrix m462(2, 2);
517    Matrix m463(2, 2);
518    Matrix m464(2, 2);
519    Matrix m465(2, 2);
520    Matrix m466(2, 2);
521    Matrix m467(2, 2);
522    Matrix m468(2, 2);
523    Matrix m469(2, 2);
524    Matrix m470(2, 2);
525    Matrix m471(2, 2);
526    Matrix m472(2, 2);
527    Matrix m473(2, 2);
528    Matrix m474(2, 2);
529    Matrix m475(2, 2);
530    Matrix m476(2, 2);
531    Matrix m477(2, 2);
532    Matrix m478(2, 2);
533    Matrix m479(2, 2);
534    Matrix m480(2, 2);
535    Matrix m481(2, 2);
536    Matrix m482(2, 2);
537    Matrix m483(2, 2);
538    Matrix m484(2, 2);
539    Matrix m485(2, 2);
540    Matrix m486(2, 2);
541    Matrix m487(2, 2);
542    Matrix m488(2, 2);
543    Matrix m489(2, 2);
544    Matrix m490(2, 2);
545    Matrix m491(2, 2);
546    Matrix m492(2, 2);
547    Matrix m493(2, 2);
548    Matrix m494(2, 2);
549    Matrix m495(2, 2);
550    Matrix m496(2, 2);
551    Matrix m497(2, 2);
552    Matrix m498(2, 2);
553    Matrix m499(2, 2);
554    Matrix m500(2, 2);
555    Matrix m501(2, 2);
556    Matrix m502(2, 2);
557    Matrix m503(2, 2);
558    Matrix m504(2, 2);
559    Matrix m505(2, 2);
560    Matrix m506(2, 2);
561    Matrix m507(2, 2);
562    Matrix m508(2, 2);
563    Matrix m509(2, 2);
564    Matrix m510(2, 2);
565    Matrix m511(2, 2);
566    Matrix m512(2, 2);
567    Matrix m513(2, 2);
568    Matrix m514(2, 2);
569    Matrix m515(2, 2);
570    Matrix m516(2, 2);
571    Matrix m517(2, 2);
572    Matrix m518(2, 2);
573    Matrix m519(2, 2);
574    Matrix m520(2, 2);
575    Matrix m521(2, 2);
576    Matrix m522(2, 2);
577    Matrix m523(2, 2);
578    Matrix m524(2, 2);
579    Matrix m525(2, 2);
580    Matrix m526(2, 2);
581    Matrix m527(2, 2);
582    Matrix m528(2, 2);
583    Matrix m529(2, 2);
584    Matrix m530(2, 2);
585    Matrix m531(2, 2);
586    Matrix m532(2, 2);
587    Matrix m533(2, 2);
588    Matrix m534(2, 2);
589    Matrix m535(2, 2);
590    Matrix m536(2, 2);
591    Matrix m537(2, 2);
592    Matrix m538(2, 2);
593    Matrix m539(2, 2);
594    Matrix m540(2, 2);
595    Matrix m541(2, 2);
596    Matrix m542(2, 2);
597    Matrix m543(2, 2);
598    Matrix m544(2, 2);
599    Matrix m545(2, 2);
600    Matrix m546(2, 2);
601    Matrix m547(2, 2);
602    Matrix m548(2, 2);
603    Matrix m549(2, 2);
604    Matrix m550(2, 2);
605    Matrix m551(2, 2);
606    Matrix m552(2, 2);
607    Matrix m553(2, 2);
608    Matrix m554(2, 2);
609    Matrix m555(2, 2);
610    Matrix m556(2, 2);
611    Matrix m557(2, 2);
612    Matrix m558(2, 2);
613    Matrix m559(2, 2);
614    Matrix m560(2, 2);
615    Matrix m561(2, 2);
616    Matrix m562(2, 2);
617    Matrix m563(2, 2);
618    Matrix m564(2, 2);
619    Matrix m565(2, 2);
620    Matrix m566(2, 2);
621    Matrix m567(2, 2);
622    Matrix m568(2, 2);
623    Matrix m569(2, 2);
624    Matrix m570(2, 2);
625    Matrix m571(2, 2);
626    Matrix m572(2, 2);
627    Matrix m573(2, 2);
628    Matrix m574(2, 2);
629    Matrix m575(2, 2);
630    Matrix m576(2, 2);
631    Matrix m577(2, 2);
632    Matrix m578(2, 2);
633    Matrix m579(2, 2);
634    Matrix m580(2, 2);
635    Matrix m581(2, 2);
636    Matrix m582(2, 2);
637    Matrix m583(2, 2);
638    Matrix m584(2, 2);
639    Matrix m585(2, 2);
640    Matrix m586(2, 2);
641    Matrix m587(2, 2);
642    Matrix m588(2, 2);
643    Matrix m589(2, 2);
644    Matrix m590(2, 2);
645    Matrix m591(2, 2);
646    Matrix m592(2, 2);
647    Matrix m593(2, 2);
648    Matrix m594(2, 2);
649    Matrix m595(2, 2);
650    Matrix m596(2, 2);
651    Matrix m597(2, 2);
652    Matrix m598(2, 2);
653    Matrix m599(2, 2);
654    Matrix m600(2, 2);
655    Matrix m601(2, 2);
656    Matrix m602(2, 2);
657    Matrix m603(2, 2);
658    Matrix m604(2, 2);
659    Matrix m605(2, 2);
660    Matrix m606(2, 2);
661    Matrix m607(2, 2);
662    Matrix m608(2, 2);
663    Matrix m609(2, 2);
664    Matrix m610(2, 2);
665    Matrix m611(2, 2);
666    Matrix m612(2, 2);
667    Matrix m613(2, 2);
668    Matrix m614(2, 2);
669    Matrix m615(2, 2);
670    Matrix m616(2, 2);
671    Matrix m617(2, 2);
672    Matrix m618(2, 2);
673    Matrix m619(2, 2);
674    Matrix m620(2, 2);
675    Matrix m621(2, 2);
676    Matrix m622(2, 2);
677    Matrix m623(2, 2);
678    Matrix m624(2, 2);
679    Matrix m625(2, 2);
680    Matrix m626(2, 2);
681    Matrix m627(2, 2);
682    Matrix m628(2, 2);
683    Matrix m629(2, 2);
684    Matrix m630(2, 2);
685    Matrix m631(2, 2);
686    Matrix m632(2, 2);
687    Matrix m633(2, 2);
688    Matrix m634(2, 2);
689    Matrix m635(2, 2);
690    Matrix m636(2, 2);
691    Matrix m637(2, 2);
692    Matrix m638(2, 2);
693    Matrix m639(2, 2);
694    Matrix m640(2, 2);
695    Matrix m641(2, 2);
696    Matrix m642(2, 2);
697    Matrix m643(2, 2);
698    Matrix m644(2, 2);
699    Matrix m645(2, 2);
700    Matrix m646(2, 2);
701    Matrix m647(2, 2);
702    Matrix m648(2, 2);
703    Matrix m649(2, 2);
704    Matrix m650(2, 2);
705    Matrix m651(2, 2);
706    Matrix m652(2, 2);
707    Matrix m653(2, 2);
708    Matrix m654(2, 2);
709    Matrix m655(2, 2);
710    Matrix m656(2, 2);
711    Matrix m657(2, 2);
712    Matrix m658(2, 2);
713    Matrix m659(2, 2);
714    Matrix m660(2, 2);
715    Matrix m661(2, 2);
716    Matrix m662(2, 2);
717    Matrix m663(2, 2);
718    Matrix m664(2, 2);
719    Matrix m665(2, 2);
720    Matrix m666(2, 2);
721    Matrix m667(2, 2);
722    Matrix m668(2, 2);
723    Matrix m669(2, 2);
724    Matrix m670(2, 2);
725    Matrix m671(2, 2);
726    Matrix m672(2, 2);
727    Matrix m673(2, 2);
728    Matrix m674(2, 2);
729    Matrix m675(2, 2);
730    Matrix m676(2, 2);
731    Matrix m677(2, 2);
732    Matrix m678(2, 2);
733    Matrix m679(2, 2);
734    Matrix m680(2, 2);
735    Matrix m681(2, 2);
736    Matrix m682(2, 2);
737    Matrix m683(2, 2);
738    Matrix m684(2, 2);
739    Matrix m685(2, 2);
740    Matrix m686(2, 2);
741    Matrix m687(2, 2);
742    Matrix m688(2, 2);
743    Matrix m689(2, 2);
744    Matrix m690(2, 2);
745    Matrix m691(2, 2);
746    Matrix m692(2, 2);
747    Matrix m693(2, 2);
748    Matrix m694(2, 2);
749    Matrix m695(2, 2);
750    Matrix m696(2, 2);
751    Matrix m697(2, 2);
752    Matrix m698(2, 2);
753    Matrix m699(2, 2);
754    Matrix m700(2, 2);
755    Matrix m701(2, 2);
756    Matrix m702(2, 2);
757    Matrix m703(2, 2);
75
```

The screenshot shows a web browser with the URL `programiz.com/cpp-programming/online-compiler/`. The page title is "Programiz C++ Online Compiler". The code editor contains the following C++ code:

```

1 #include <iostream>
2 #include <cmath> // For M_PI
3 double calculateArea(double length, double width) {
4     return length * width;
5 }
6 double calculateArea(double radius) {
7     return M_PI * radius * radius;
8 }
9 double calculateArea(double base, int height) {
10    return 0.5 * base * height;
11 }
12 int main() {
13     double length, width, radius, base;
14     int height;
15     std::cout << "Enter the length and width of the rectangle: ";
16     std::cin >> length >> width;
17     std::cout << "Area of the rectangle: " << calculateArea(length, width) << std::endl;
18     std::cout << "Enter the radius of the circle: ";
19     std::cin >> radius;
20     std::cout << "Area of the circle: " << calculateArea(radius) << std::endl;
21     std::cout << "Enter the base and height of the triangle: ";
22     std::cin >> base >> height;
23     std::cout << "Area of the triangle: " << calculateArea(base, height) << std::endl;
24     return 0;

```

The output window shows the following text:

```

/tmp/6WzZcp1292.o
Enter the length and width of the rectangle: 7
8
Area of the rectangle: 56
Enter the radius of the circle: 7
Area of the circle: 153.938
Enter the base and height of the triangle: 9
4
Area of the triangle: 18

```

- Develop a C++ program that evaluates mathematical expressions entered by the user, considering operator precedence and associativity. Support parentheses for grouping.

The screenshot shows a web browser with the URL `programiz.com/cpp-programming/online-compiler/`. The page title is "Programiz C++ Online Compiler". The code editor contains the following C++ code:

```

1 #include <iostream>
2 #include <stack>
3 #include <sstream>
4 #include <cctype>
5 #include <map>
6 #include <stdexcept>
7 bool isOperator(char c) {
8     return c == '+' || c == '-' || c == '*' || c == '/';
9 }
10 int precedence(char op) {
11     if (op == '+' || op == '-') return 1;
12     if (op == '*' || op == '/') return 2;
13     return 0;
14 }
15 int applyOp(int a, int b, char op) {
16     switch (op) {
17         case '+': return a + b;
18         case '-': return a - b;
19         case '*': return a * b;
20         case '/':
21             if (b == 0) throw std::runtime_error("Division by zero.");
22             return a / b;
23     }
24     return 0;
25 }
26 std::string infixToPostfix(const std::string& infix) {

```

The output window shows the following text:

```

/tmp/y3F7usulwb.o
Enter a mathematical expression: (8+9)*6-5
The result is: 97

```

- Write a C++ program that defines a template function to find the maximum element in an array of any data type. Test the function with integer, float, and double arrays.

The screenshot shows a web browser with the URL `programiz.com/cpp-programming/online-compiler/`. The page title is "Programiz C++ Online Compiler". There is a "C++ Certification" button in the top right. The main area is divided into two panels: "main.cpp" on the left and "Output" on the right. The "main.cpp" panel contains the following code:

```
1 #include <iostream>
2 template<typename T>
3 T findMax(T arr[], int size) {
4     T max = arr[0];
5     for (int i = 1; i < size; ++i) {
6         if (arr[i] > max) {
7             max = arr[i];
8         }
9     }
10    return max;
11 }
12 int main() {
13     int intArray[] = {1, 2, 3, 4, 5};
14     std::cout << "Maximum element in integer array: " << findMax(intArray, 5) <<
        std::endl;
15     float floatArray[] = {1.1f, 2.2f, 3.3f, 4.4f, 5.5f};
16     std::cout << "Maximum element in float array: " << findMax(floatArray, 5) <<
        std::endl;
17     double doubleArray[] = {1.01, 2.02, 3.03, 4.04, 5.05};
18     std::cout << "Maximum element in double array: " << findMax(doubleArray, 5)
        << std::endl;
19     return 0;
20 }
```

The "Output" panel shows the following text:

```
/tmp/myN03v3CK9.o
Maximum element in integer array: 5
Maximum element in float array: 5.5
Maximum element in double array: 5.05
```

7. Create a C++ program to implement a recursive function to solve the Tower of Hanoi puzzle with 'n' disks.

The screenshot shows the same online C++ compiler interface. The "main.cpp" panel contains the following code:

```
1 #include <iostream>
2 void moveDisk(int n, char fromPeg, char toPeg, char auxPeg) {
3     if (n == 1) {
4         std::cout << "Move disk 1 from peg " << fromPeg << " to peg " << toPeg <<
            std::endl;
5         return;
6     }
7     moveDisk(n - 1, fromPeg, auxPeg, toPeg);
8     std::cout << "Move disk " << n << " from peg " << fromPeg << " to peg " <<
        toPeg << std::endl;
9     moveDisk(n - 1, auxPeg, toPeg, fromPeg);
10 }
11 int main() {
12     int n; // Number of disks
13     std::cout << "Enter the number of disks: ";
14     std::cin >> n;
15     moveDisk(n, 'A', 'C', 'B'); // A, B and C are names of the pegs
16     return 0;
17 }
```

The "Output" panel shows the following text:

```
/tmp/0oBj0zneCV.o
Enter the number of disks: 3
Move disk 1 from peg A to peg C
Move disk 2 from peg A to peg B
Move disk 1 from peg C to peg B
Move disk 3 from peg A to peg C
Move disk 1 from peg B to peg A
Move disk 2 from peg B to peg C
Move disk 1 from peg A to peg C
```

8. Develop a C++ program that reads input from a file containing numeric data and calculates the average. Implement exception handling to handle file I/O errors and invalid data.

The screenshot shows a web browser with the URL `programiz.com/cpp-programming/online-compiler/`. The page title is "Programiz C++ Online Compiler". A "C++ Certification" button is visible in the top right. The main area contains a code editor with a file named `main.cpp` and an "Output" panel on the right.

```

1 #include <iostream>
2 #include <fstream>
3 #include <string>
4 #include <stdexcept>
5 int main() {
6     std::string filename;
7     std::cout << "Enter the filename: ";
8     std::cin >> filename;
9     std::ifstream inputFile(filename);
10    double number, sum = 0.0;
11    int count = 0;
12    if (!inputFile.is_open()) {
13        std::cerr << "Error opening file: " << filename << std::endl;
14        return 1;
15    }
16    while (inputFile >> number) {
17        sum += number;
18        ++count;
19    }
20    if (!inputFile.eof()) {
21        std::cerr << "Error reading data from file." << std::endl;
22        inputFile.close();
23        return 1;
24    }
25    inputFile.close();
26    if (count == 0) {

```

The output panel shows the following text:

```

/tmp/ITZbvbiGQY.o
Enter the filename: 27question.txt
Error opening file: 27question.txt

```

9. Write a C++ program that implements a callback mechanism using function pointers. Create a function that takes a callback function as an argument and calls it.

The screenshot shows the same online C++ compiler interface. The code editor now contains a program that demonstrates a callback mechanism using function pointers.

```

1 #include <iostream>
2 typedef void (*CallbackFunction)(int);
3 void myCallback(int data)
4 {
5     std::cout << "Callback called with data: " << data << std::endl;
6 }
7 void process(int data, CallbackFunction callback)
8 {
9     std::cout << "Processing data: " << data << std::endl;
10    callback(data);
11 }
12 int main()
13 {
14     int someData = 10;
15     process(someData, myCallback);
16     return 0;
17 }
18

```

The output panel shows the following text:

```

/tmp/F9DgLhD0pR.o
Processing data: 10
Callback called with data: 10

```

10. Develop a C++ program to implement a binary search tree (BST) and perform operations such as insertion, deletion, and traversal (in-order, pre-order, post-order).

Programiz
C++ Online Compiler

C++ Certification

main.cpp

```
1 #include <iostream>
2
3 struct TreeNode {
4     int value;
5     TreeNode *left, *right;
6
7     TreeNode(int val) : value(val), left(nullptr), right(nullptr) {}
8 };
9
10 class BST {
11 public:
12     TreeNode *root;
13
14     BST() : root(nullptr) {}
15
16     void insert(int val) {
17         root = insertRec(root, val);
18     }
19
20     void remove(int val) {
21         root = removeRec(root, val);
22     }
23     void inOrder() {
24         inOrderRec(root);
25         std::cout << std::endl;
26     }
27 }
```

Run

Output

Clear

```
/tmp/308jV3pvk6.o
In-order traversal: 1 3 4 6 7 8 10 13 14
Pre-order traversal: 8 3 1 6 4 7 10 14 13
Post-order traversal: 1 4 7 6 3 13 14 10 8
After deleting 10, in-order traversal: 1 3 4 6 7 8 13 14
```