main.cpp

Run

```cpp
#include <iostream>
#include <cmath>
using namespace std;
bool isPrime(int num) {
    if (num <= 1) {
        return false;
    }
    for (int i = 2; i <= sqrt(num); i++) {
        if (num % i == 0) {
            return false;
        }
    }
    return true;
}
int main() {
    int number;
    cout << "Enter a number: ";
    cin >> number;
    if (isPrime(number)) {
        cout << number << " is a prime number." << endl;
    } else {
        cout << number << " is not a prime number." << endl;
    }
    return 0;
}
```

Output

Clear

```
/tmp/hc5CUk0C48.o
Enter a number: 2
2 is a prime number.
```

main.cpp

Run

```cpp
#include <iostream>
#include <string>
#include <algorithm>

class StringManipulation {
public:
    StringManipulation() {}

    void countCapitals(const std::string& str) {
        int capitalCount = std::count_if(str.begin(), str.end(), [](char c)
            {
            return std::isupper(c);
        });
        std::cout << "Number of capital letters: " << capitalCount << std
            ::endl;
    }
};

class StringConverter {
public:
    StringConverter() {}

    std::string convertToUpper(const std::string& str) {
        std::string upperStr;
        std::transform(str.begin(), str.end(), std::back_inserter(upperStr
```

Output

Clear

```
/tmp/hc5CUkOC48.o
Enter a string: Chinnu
Original String: Chinnu
Number of capital letters: 1
Uppercase String: CHINNU
Lowercase String: chinnu
```

main.cpp

Run

```cpp
#include <iostream>

class CubeCalculator {
public:

    CubeCalculator() {

    }
    double calculateCube(double num) {
        return num * num * num;
    }
};

int main() {

    CubeCalculator cubeCalculator;

    double number;
    std::cout << "Enter a number: ";
    std::cin >> number;

    double result = cubeCalculator.calculateCube(number);
    std::cout << "Cube of " << number << " is: " << result << std::endl;

    return 0;
```

Output

Clear

```
/tmp/b5rbkekC6Z.o
Enter a number: 3
Cube of 3 is: 27
```

main.cpp

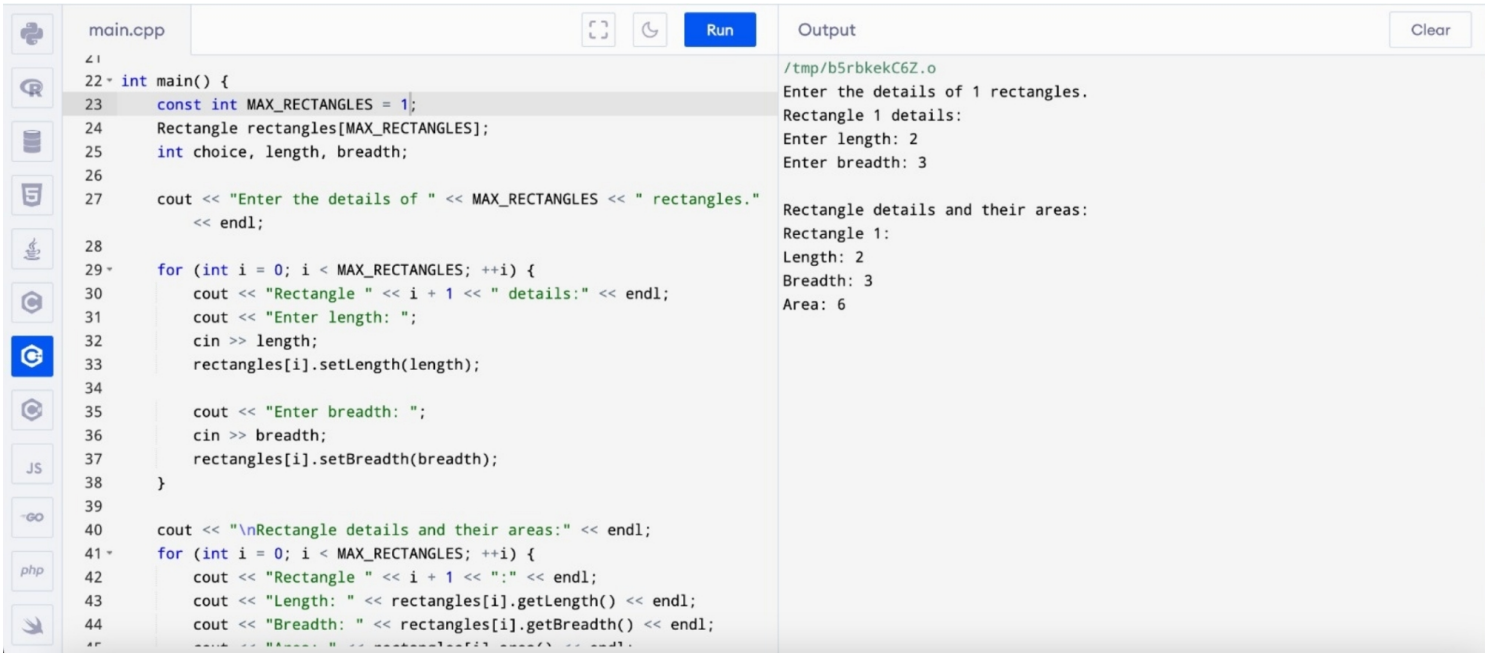
Run

```cpp
#include <iostream>

int main() {
    int n, i, sum = 0;

    std::cout << "Enter a positive integer: ";
    std::cin >> n;

    for (i = 1; i <= n; ++i) {
        sum = sum + i;
    }

    std::cout << "Sum of first " << n << " natural numbers: " << sum << std
        ::endl;

    return 0;
}
```

Output

Clear

```
/tmp/hc5CUk0C48.o
Enter a positive integer: 5
Sum of first 5 natural numbers: 15
```


main.cpp
Run
Output
Clear
int main() {
const int MAX_RECTANGLES = 1;
Rectangle rectangles[MAX_RECTANGLES];
int choice, length, breadth;
cout << "Enter the details of " << MAX_RECTANGLES << " rectangles."
<< endl;
for (int i = 0; i < MAX_RECTANGLES; ++i) {
cout << "Rectangle " << i + 1 << " details:" << endl;
cout << "Enter length: ";
cin >> length;
rectangles[i].setLength(length);
cout << "Enter breadth: ";
cin >> breadth;
rectangles[i].setBreadth(breadth);
}
cout << "\nRectangle details and their areas:" << endl;
for (int i = 0; i < MAX_RECTANGLES; ++i) {
cout << "Rectangle " << i + 1 << ":" << endl;
cout << "Length: " << rectangles[i].getLength() << endl;
cout << "Breadth: " << rectangles[i].getBreadth() << endl;
/tmp/b5rbkekC6Z.o
Enter the details of 1 rectangles.
Rectangle 1 details:
Enter length: 2
Enter breadth: 3
Rectangle details and their areas:
Rectangle 1:
Length: 2
Breadth: 3
Area: 6

main.cpp

Run

```
13      cout << "Enter the operation (+, -, *, /, %): ";
14      cin >> operation;
15
16      switch (operation) {
17          case '+':
18              cout << "Result: " << num1 + num2 << endl;
19              break;
20          case '-':
21              cout << "Result: " << num1 - num2 << endl;
22              break;
23          case '*':
24              cout << "Result: " << num1 * num2 << endl;
25              break;
26          case '/':
27              if (num2 != 0)
28                  cout << "Result: " << num1 / num2 << endl;
29              else
30                  cout << "Error! Division by zero is not allowed." <<
                        endl;
31              break;
32          case '%':
33              cout << "Result: " << static_cast<int>(num1) % static_cast
                    <int>(num2) << endl;
34              break;
35          default:
```

Output

Clear

```
/tmp/b5rbkekC6Z.o
Enter the first number: 2
Enter the second number: 3
Enter the operation (+, -, *, /, %): +
Result: 5
```

main.cpp

Run

```
    PatternPrinter(int num) : num_(num) {}

    void printPattern() {
        for (int i = 1; i <= num_; ++i) {
            for (int j = 1; j <= i; ++j) {
                cout << num_ << " ";
            }
            cout << endl;
        }
    }

private:
    int num_;
};

int main() {
    int num;
    cout << "Enter a number to repeat in the pattern: ";
    cin >> num;

    PatternPrinter pattern(num);
    pattern.printPattern();

    return 0;
}
```

Output

Clear

```
/tmp/b5rbkekC6Z.o
Enter a number to repeat in the pattern: 4
4
4 4
4 4 4
4 4 4 4
```

main.cpp

Run

```cpp
#include <iostream>
#include <vector>
#include <queue>

std::vector<int> mergeArrays(const std::vector<std::vector<int>>& lists) {
    std::vector<int> result;

    std::priority_queue<std::pair<int, std::pair<int, int>>, std::vector
        <std::pair<int, std::pair<int, int>>>, std::greater<>> minHeap;

    for (int i = 0; i < lists.size(); ++i) {
        if (!lists[i].empty()) {
            minHeap.push({lists[i][0], {i, 0}});
        }
    }

    while (!minHeap.empty()) {

        auto top = minHeap.top();
        minHeap.pop();

        result.push_back(top.first);

        int listIndex = top.second.first;
        int nextIndex = top.second.second + 1;
```

Output

Clear

```
/tmp/b5rbkekC6Z.o
Output: [1, 1, 2, 3, 4, 4, 5, 6]
```