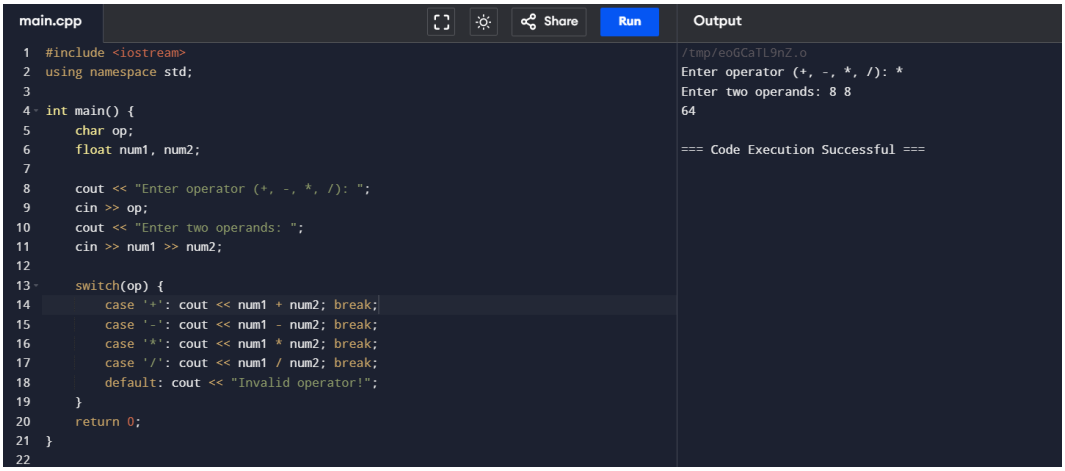
Complex Problems

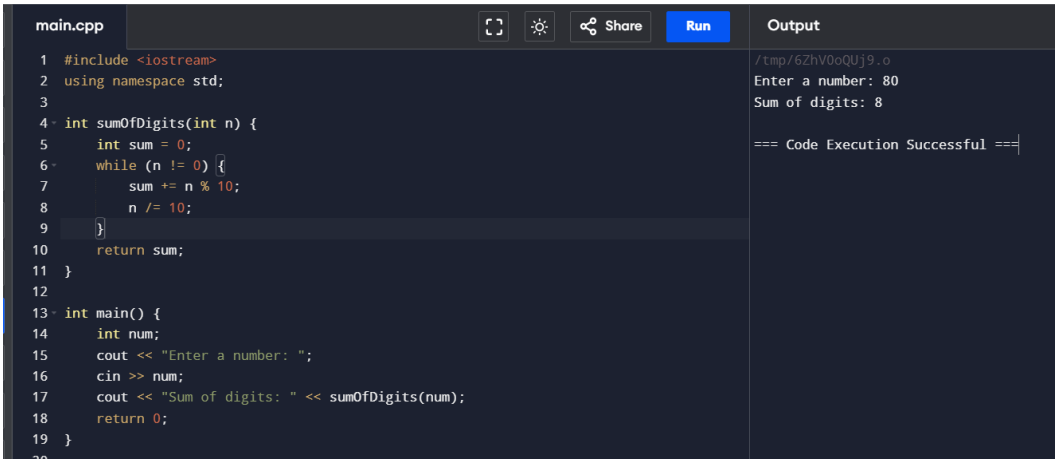
Problem 1: Simple Calculator

Write a basic calculator program that performs addition, subtraction, multiplication, and division.



Problem 2: Sum of Digits

Write a program to find the sum of the digits of a given number.



Problem 3: Sum of First N Natural Numbers

Develop a program that calculates the sum of the first N natural numbers.

main.cpp

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

int main() {
    int n;
    cout << "Enter a number: ";
    cin >> n;
    cout << "Sum of first " << n << " natural numbers: " << (n * (n + 1)) / 2;
    return 0;
}
```

Output

```
/tmp/m2scnZtk7W.o
Enter a number: 89
Sum of first 89 natural numbers: 4005

=== Code Execution Successful ===
```

Problem 4: Count Vowels in a String

Write a program that counts the number of vowels in a given string.

main.cpp

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

int countVowels(const string &str) {
    int count = 0;
    for (char c : str) {
        if (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u' ||
            c == 'A' || c == 'E' || c == 'I' || c == 'O' || c == 'U') {
            count++;
        }
    }
    return count;
}

int main() {
    string str;
    cout << "Enter a string: ";
    getline(cin, str);
    cout << "Vowel count: " << countVowels(str);
    return 0;
}
```

Output

```
/tmp/8SlJm7CEc9.o
Enter a string: waqas zafar
Vowel count: 4

=== Code Execution Successful ===
```

Problem 5: Simple Interest Calculator

Create a program to calculate simple interest based on principal, rate, and time.

main.cpp

```cpp
using namespace std;

int main() {
    float principal, rate, time;
    cout << "Enter the principal amount (initial loan or investment): ";
    cin >> principal;
    cout << "Enter the annual interest rate (as a percentage, e.g., 5 for 5%): ";
    cin >> rate;
    cout << "Enter the time period (in years): ";
    cin >> time;
    float simpleInterest = (principal * rate * time) / 100;

    float totalAmount = principal + simpleInterest;

    cout << "\n--- Interest Calculation ---\n";
    cout << "Principal Amount: PKR" << principal << endl;
    cout << "Annual Interest Rate: " << rate << "%" << endl;
    cout << "Time Period: " << time << " years" << endl;
    cout << "Simple Interest: $" << simpleInterest << endl;
    cout << "Total Amount after " << time << " years: $" << totalAmount << endl;

    return 0;
}
```

Output

```
/tmp/9UOlYYqF3U.o
Enter the principal amount (initial loan or investment): 100000
Enter the annual interest rate (as a percentage, e.g., 5 for 5%): 2
Enter the time period (in years): 3

--- Interest Calculation ---
Principal Amount: PKR100000
Annual Interest Rate: 2%
Time Period: 3 years
Simple Interest: $6000
Total Amount after 3 years: $106000


=== Code Execution Successful ===
```

Problem 6: Multiplication Table Generator

Write a program to generate and display the multiplication table of any given number.

Problem 7: Count Positive and Negative Numbers

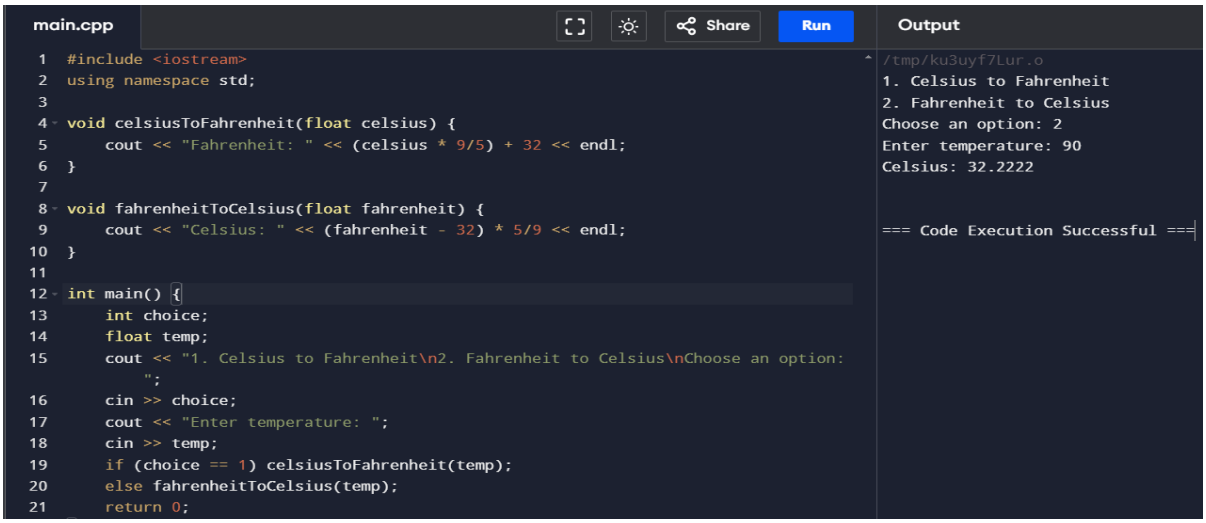
Write a program that reads a list of numbers and counts how many are positive, negative, and zero.

Problem 8: Leap Year Checker

Write a program that checks if a given year is a leap year or not.

Problem 9: Temperature Converter

Write a program that converts temperatures from Celsius to Fahrenheit and vice versa.



Problem 10: Swap Two Numbers

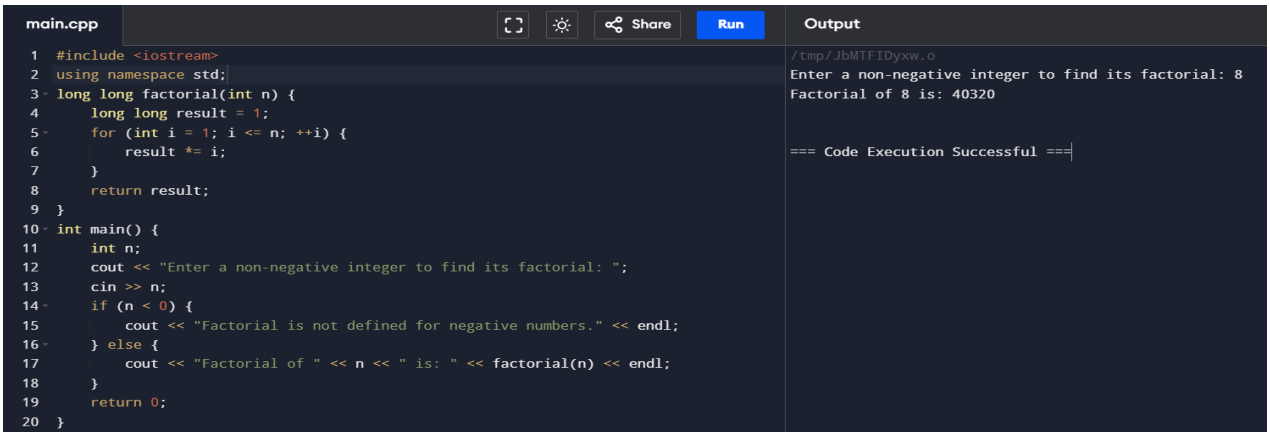
Write a program that swap two number.

Problem 11: Add Two numbers without the + operator.

Write a program that Add Two numbers without + operator.

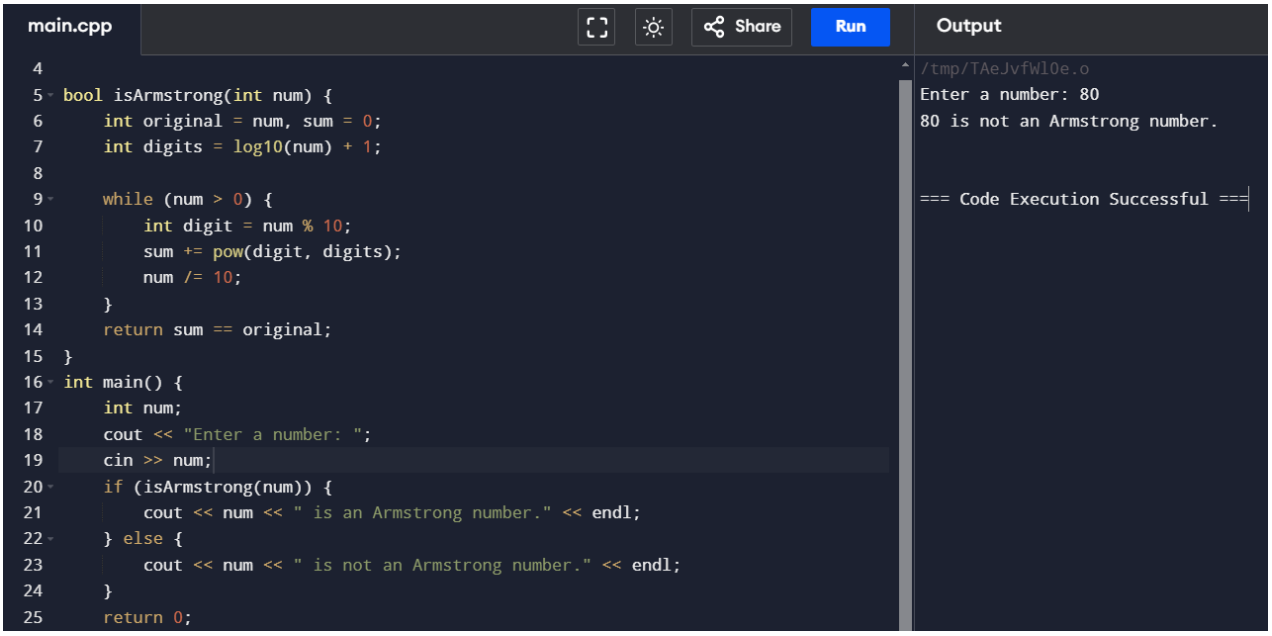
Problem 12: Factorial Calculation.

Write an algorithm to calculate the factorial of a given positive integer using iteration.



Problem 19: Armstrong Number

Write a program to check if a given number is an Armstrong number (a number that is equal to the sum of its own digits raised to the power of the number of digits).



Problem 20: Find the Second Largest Number

Write a program that finds the second largest number in an array of integers.

main.cpp

Share Run

Output

```cpp
int findSecondLargest(int arr[], int n) {
    int first = INT_MIN, second = INT_MIN;

    for (int i = 0; i < n; ++i) {
        if (arr[i] > first) {
            second = first;
            first = arr[i];
        } else if (arr[i] > second && arr[i] < first) {
            second = arr[i];
        }
    }
    return second;
}
int main() {
    int arr[] = {12, 35, 1, 10, 34, 1};
    int n = sizeof(arr) / sizeof(arr[0]);

    int result = findSecondLargest(arr, n);
    if (result != INT_MIN) {
        cout << "The second largest element is: " << result << endl;
    } else {
        cout << "No second largest element found." << endl;
    }
```

```
/tmp/RUlxVLJLKn.o
The second largest element is: 34


=== Code Execution Successful ===
```

Problem 21: Second Smallest Number

Write a program to find the second smallest number in an array of integers.

main.cpp

Share Run

Output

```cpp
using namespace std;

int findSecondSmallest(int arr[], int n) {
    int first = INT_MAX, second = INT_MAX;
    for (int i = 0; i < n; ++i) {
        if (arr[i] < first) {
            second = first;
            first = arr[i];
        } else if (arr[i] < second && arr[i] > first) {
            second = arr[i];
        }
    }
    return second;
}
int main() {
    int arr[] = {12, 35, 1, 10, 34, 1};
    int n = sizeof(arr) / sizeof(arr[0]);
    int result = findSecondSmallest(arr, n);
    if (result != INT_MAX) {
        cout << "The second smallest element is: " << result << endl;
    } else {
        cout << "No second smallest element found." << endl;
    }
    return 0;
```

```
/tmp/L1LCVsR2pr.o
The second smallest element is: 10


=== Code Execution Successful ===
```

Problem 22: Second Largest in a Sorted Array

Given a sorted array, write a program to find the second largest number without iterating through the entire array.

main.cpp

Share Run

Output

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

int findSecondLargestSorted(int arr[], int n) {
    if (n < 2) return -1;
    int largest = arr[n - 1];
    for (int i = n - 2; i >= 0; --i) {
        if (arr[i] < largest) {
            return arr[i];
        }
    }
    return -1;
}
int main() {
    int arr[] = {1, 2, 3, 4, 5, 5};
    int n = sizeof(arr) / sizeof(arr[0]);
    int result = findSecondLargestSorted(arr, n);
    if (result != -1) {
        cout << "The second largest element is: " << result << endl;
    } else {
        cout << "No second largest element found." << endl;
    }
    return 0;
```

```
/tmp/DqHFQLpUCZ.o
The second largest element is: 4


=== Code Execution Successful ===
```

Problem 23: Remove Duplicates and Find Second Largest

Write a program that removes duplicate numbers from an array and then finds the second largest number in the resulting array.

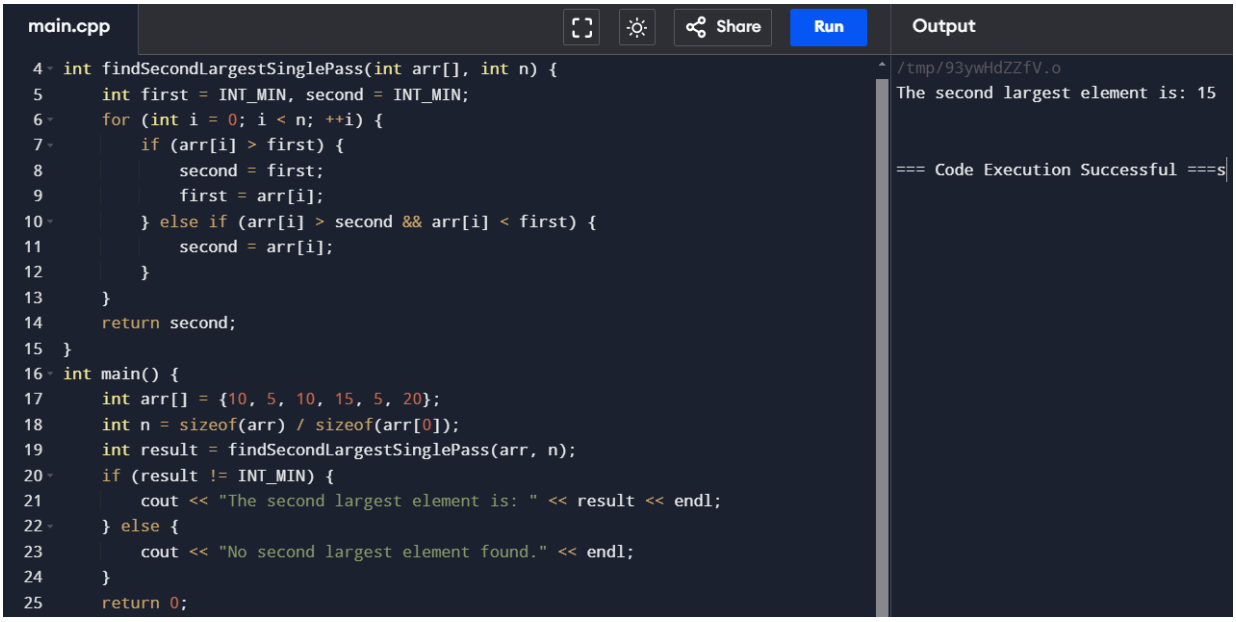
```cpp
int removeDuplicates(int arr[], int n) {
    if (n == 0 || n == 1) return n;
    int temp[n];
    int j = 0;
    for (int i = 0; i < n - 1; ++i) {
        if (arr[i] != arr[i + 1]) {
            temp[j++] = arr[i];
        }
    }
    temp[j++] = arr[n - 1];
    for (int i = 0; i < j; ++i) {
        arr[i] = temp[i];
    }
    return j;
}
int findSecondLargestAfterRemovingDuplicates(int arr[], int n) {
    sort(arr, arr + n);
    int newSize = removeDuplicates(arr, n);
    if (newSize < 2) return -1;
    return arr[newSize - 2];
}
int main() {
    int arr[] = {1, 2, 3, 4, 5, 5, 3};
    int n = sizeof(arr) / sizeof(arr[0]);
    int result = findSecondLargestAfterRemovingDuplicates(arr, n);
    if (result != -1) {
        cout << "The second largest element after:  " << result ;
```

```
Output:

The second largest element after:  4
```

Problem 24: Second Largest Using a Single Pass

Write an algorithm to find the second largest number in an array in a single pass (without sorting the array).




```cpp
int findSecondLargestSinglePass(int arr[], int n) {
    int first = INT_MIN, second = INT_MIN;
    for (int i = 0; i < n; ++i) {
        if (arr[i] > first) {
            second = first;
            first = arr[i];
        } else if (arr[i] > second && arr[i] < first) {
            second = arr[i];
        }
    }
    return second;
}
int main() {
    int arr[] = {10, 5, 10, 15, 5, 20};
    int n = sizeof(arr) / sizeof(arr[0]);
    int result = findSecondLargestSinglePass(arr, n);
    if (result != INT_MIN) {
        cout << "The second largest element is: " << result << endl;
    } else {
        cout << "No second largest element found." << endl;
    }
    return 0;
```

Problem 25: Second Largest in a 2D Array

Write a program to find the second largest number in a 2D array of integers.


```cpp
int findSecondLargest2D(int arr[][3], int rows, int cols) {
    int first = INT_MIN, second = INT_MIN;

    for (int i = 0; i < rows; ++i) {
        for (int j = 0; j < cols; ++j) {
            if (arr[i][j] > first) {
                second = first;
                first = arr[i][j];
            } else if (arr[i][j] > second && arr[i][j] < first) {
                second = arr[i][j];
            }
        }
    }
    return second;
}
int main() {
    int arr[3][3] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};
    int rows = 3, cols = 3;
    int result = findSecondLargest2D(arr, rows, cols);
    if (result != INT_MIN) {
        cout << "The second largest element in the 2D array is: " << result <<
            endl;
    if (result != INT_MIN) {
        cout << "The second largest element in the 2D array is: " << result <<
            endl;
    } else {
        cout << "No second largest element found." << endl;
    }
    return 0;
```