

Sum of n natural numbers

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

int main() {
    // Declare a variable to store the input value
    int n;

    // input from user
    cout << "Enter a positive integer: ";
    cin >> n;

    // calculate sum
    int sum = n * (n + 1) / 2;

    // Output the result
    cout << "The sum of natural numbers from 1 to " << n << " is: " << sum << endl;

    return 0;
}
```

```
Enter a positive integer: 2
The sum of natural numbers from 1 to 2 is: 3
```

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

1) Check if a Number is Prime

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

bool isPrime(int n) {
    // Handle base cases
    if (n <= 1) return false; // 0 and 1 are not prime numbers
    if (n == 2) return true; // 2 is a prime number

    // Check for divisibility from 2 to sqrt(n)
    for (int i = 2; i <= sqrt(n); i++) {
        if (n % i == 0) {
            return false; // If n is divisible by i, it is not prime
        }
    }

    return true; // n is prime
}

int main() {
```

```
int n;
```

```

// Take input from the user
cout << "Enter a number: ";
cin >> n;

// Check if the number is prime
if (isPrime(n)) {
    cout << n << " is a prime number." << endl;
} else {
    cout << n << " is not a prime number." << endl;
}

return 0;
}

```

Output

```

Enter a number: 5
5 is a prime number.

```

```

=== Code Execution Successful ===

```

2) Print Odd Numbers up to N

```

#include <iostream>
using namespace std;

int main() {
    int n;

    // Take input from the user
    cout << "Enter a number: ";
    cin >> n;

    // Print all odd numbers between 1 and n (inclusive)
    for (int i = 1; i <= n; i++) {
        if (i % 2 != 0) { // Check if the number is odd
            cout << i << " "; // Print the odd number
        }
    }

    cout << endl; // Print a newline at the end
    return 0;
}

```

Output

Enter a number: 2

1

=== Code Execution Successful ===

3) Print Multiplication Table of a Number

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {
```

```
    int n;
```

```
    // Take input from the user
```

```
    cout << "Enter a number: ";
```

```
    cin >> n;
```

```
    // Print the multiplication table of n
```

```
    for (int i = 1; i <= 10; i++) {
```

```
        cout << n << " × " << i << " = " << n * i << endl;
```

```
    }
```

```
    return 0;
```

```
}
```

Output

Enter a number: 5

5 × 1 = 5

5 × 2 = 10

5 × 3 = 15

5 × 4 = 20

5 × 5 = 25

5 × 6 = 30

5 × 7 = 35

5 × 8 = 40

5 × 9 = 45

5 × 10 = 50

4) Count Digits in a Number

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {
```

```
    int n;
```

```

// Take input from the user
cout << "Enter a number: ";
cin >> n;

// Initialize a counter for the number of digits
int digitCount = 0;

// Handle the special case when n is 0
if (n == 0) {
    digitCount = 1; // 0 has 1 digit
} else {
    // Count the digits using a loop
    while (n != 0) {
        n /= 10; // Divide n by 10, effectively removing the last digit
        digitCount++; // Increment the digit count
    }
}

// Output the total number of digits
cout << "The number of digits is: " << digitCount << endl;

return 0;
}

```

Output

```

Enter a number: 4
The number of digits is: 1

```

```

=== Code Execution Successful ===

```

5) Find the Sum of Digits of a Number

```

#include <iostream>
using namespace std;

int main() {
    int n;

    // Take input from the user
    cout << "Enter a number: ";
    cin >> n;

    int sum = 0;

    // Handle the special case where n is 0
    if (n == 0) {
        sum = 0; // Sum of digits for 0 is 0
    }
}

```

```

    } else {
        // Calculate the sum of digits
        while (n != 0) {
            sum += n % 10; // Add the last digit to sum
            n /= 10;       // Remove the last digit
        }
    }

    // Output the sum of digits
    cout << "The sum of the digits is: " << sum << endl;

    return 0;
}

```

Output

```

Enter a number: 4
The sum of the digits is: 4

=== Code Execution Successful ===

```

6) Reverse a Number

```

#include <iostream>
using namespace std;

int main() {
    int n;

    // Take input from the user
    cout << "Enter a number: ";
    cin >> n;

    int reversed = 0;

    // Handle the special case where n is 0
    if (n == 0) {
        reversed = 0;
    } else {
        // Reverse the digits of the number
        while (n != 0) {
            int digit = n % 10; // Extract the last digit
            reversed = reversed * 10 + digit; // Add the digit to the reversed number
            n /= 10;           // Remove the last digit from n
        }
    }

    // Output the reversed number
    cout << "The reversed number is: " << reversed << endl;

    return 0;
}

```

Output

```
Enter a number: 5
The reversed number is: 5
```

7) Find the Largest Digit in a Number

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {
    int n;
```

```
    // Take input from the user
```

```
    cout << "Enter a number: ";
```

```
    cin >> n;
```

```
    int largestDigit = -1; // Initialize largestDigit to a value smaller than any digit (0-9)
```

```
    // Handle the case where n is 0 (the largest digit is 0)
```

```
    if (n == 0) {
```

```
        largestDigit = 0;
```

```
    } else {
```

```
        // Loop to extract each digit and find the largest one
```

```
        while (n != 0) {
```

```
            int digit = n % 10; // Extract the last digit
```

```
            if (digit > largestDigit) {
```

```
                largestDigit = digit; // Update largestDigit if the current digit is larger
```

```
            }
```

```
            n /= 10; // Remove the last digit
```

```
        }
```

```
    }
```

```
    // Output the largest digit
```

```
    cout << "The largest digit is: " << largestDigit << endl;
```

```
    return 0;
```

```
}
```

Output

```
Enter a number: 4
The largest digit is: 4
```

8) Check if a Number is a Palindrome

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {
```

```
    int n, originalNumber, reversedNumber = 0, remainder;
```

```
    // Take input from the user
```

```
    cout << "Enter a number: ";
```

```
    cin >> n;
```

```

originalNumber = n; // Store the original number to compare later

// Reverse the digits of the number
while (n != 0) {
    remainder = n % 10;          // Extract the last digit
    reversedNumber = reversedNumber * 10 + remainder; // Build the reversed number
    n /= 10;                     // Remove the last digit
}

// Compare the original number with the reversed number
if (originalNumber == reversedNumber) {
    cout << originalNumber << " is a palindrome." << endl;
} else {
    cout << originalNumber << " is not a palindrome." << endl;
}

return 0;
}

```

Output

```

Enter a number: 9
9 is a palindrome.

```

```

=== Code Execution Successful ===

```

9) Implement Multiple Inheritance to Simulate a Library System

```

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

// Base class to store book details
class Book {
public:
    string title;
    string author;
    string ISBN;

    // Constructor to initialize book details
    Book(string t, string a, string isbn) : title(t), author(a), ISBN(isbn) {}

    // Function to display book details
    void displayBookDetails() {
        cout << "Title: " << title << endl;
        cout << "Author: " << author << endl;
        cout << "ISBN: " << ISBN << endl;
    }
};

// Base class to store borrower details
class Borrower {
public:
    string name;

```



```
int id;
string borrowedBook;

// Constructor to initialize borrower details
Borrower(string n, int i) : name(n), id(i), borrowedBook("") {}

// Function to borrow a book
void borrowBook(Book &book) {
    if (borrowedBook == "") {
        borrowedBook = book.title;
        cout << name << " has borrowed the book: " << borrowedBook << endl;
    } else {
        cout << name << " already has a book borrowed: " << borrowedBook << endl;
    }
}
```