1. GCD number

```
#include <iostream>

int gcd(int a, int b) {
    while (b != 0) {
        int temp = b;
        b = a % b;
        a = temp;
    }
    return a;
}

int main() {
    int a = 56;
    int b = 98;
    std::cout << "GCD of " << a << " and " << b << " is " << gcd(a, b) << std::endl;
    return 0;
}
```

2.linklist reversal

```
#include <iostream>

struct ListNode {
    int val;
    ListNode* next;
    ListNode(int x) : val(x), next(nullptr) {}
};

ListNode* reverseList(ListNode* head) {
    ListNode* prev = nullptr;
    ListNode* current = head;
    ListNode* next = nullptr;

    while (current != nullptr) {

        next = current->next;

        current->next = prev;

        prev = current;
        current = next;
    }

    return prev;
}

void printList(ListNode* head) {
    ListNode* current = head;
    while (current != nullptr) {
        std::cout << current->val << " ";
        current = current->next;
    }
    std::cout << std::endl;
}

int main() {
    ListNode* head = new ListNode(1);
    head->next = new ListNode(2);
    head->next->next = new ListNode(3);
    head->next->next->next = new ListNode(4);
    head->next->next->next->next = new ListNode(5);

    std::cout << "Original list: ";
    printList(head);

    head = reverseList(head);

    std::cout << "Reversed list: ";
    printList(head);


    ListNode* current = head;
    while (current != nullptr) {
        ListNode* next = current->next;
        delete current;
        current = next;
    }

    return 0;
}
```

3.Perfect number

```
#include <iostream>

bool isPerfectNumber(int num) {
    if (num <= 1) return false;
    int sum = 0;

    for (int i = 1; i <= num / 2; ++i) {
        if (num % i == 0) {
            sum += i;
        }
    }

    return sum == num;
}

int main() {
    int num = 28;
    if (isPerfectNumber(num)) {
        std::cout << num << " is a perfect number." << std::endl;
    } else {
        std::cout << num << " is not a perfect number." << std::endl;
    }
    return 0;
}
```

4.Prime Number or not

```
#include <iostream>
#include <cmath>

bool isPrime(int n) {
    if (n <= 1) return false;
    for (int i = 2; i <= std::sqrt(n); ++i) {
        if (n % i == 0) return false;
    }
    return true;
}

int main() {
    int num = 29;
    if (isPrime(num)) {
        std::cout << num << " is a prime number." << std::endl;
    } else {
        std::cout << num << " is not a prime number." << std::endl;
    }
    return 0;
}
```

5.string Reverse

```
#include <iostream>
#include <string>

std::string reverseString(const std::string& str) {
    int n = str.length();
    std::string reversedStr = str;
    for (int i = 0; i < n / 2; ++i) {
        std::swap(reversedStr[i], reversedStr[n - i - 1]);
    }
    return reversedStr;
}

int main() {
    std::string str = "hello";
    std::string reversedStr = reverseString(str);
    std::cout << "Original string: " << str << std::endl;
    std::cout << "Reversed string: " << reversedStr << std::endl;
    return 0;
}
```

6.Array Sum

```
#include <iostream>
#include <numeric>

int arraySum(int arr[], int size) {
    return std::accumulate(arr, arr + size, 0);
}

int main() {
    int arr[] = {1, 2, 3, 4, 5};
    int size = sizeof(arr) / sizeof(arr[0]);
    std::cout << "Sum: " << arraySum(arr, size) << std::endl;
    return 0;
}
```