

```

#include <iostream>
#include <vector>

int countPrimesNonRecursive(int n) {
    if (n < 2) return 0;

    // Создаем вектор и заполняем его значениями true
    std::vector<bool> is_prime(n + 1, true);
    is_prime[0] = is_prime[1] = false;

    // Алгоритм "Решето Эратосфена"
    for (int p = 2; p * p <= n; p++) {
        if (is_prime[p]) {
            for (int i = p * p; i <= n; i += p)
                is_prime[i] = false;
        }
    }

    // Считаем количество оставшихся простых чисел
    int count = 0;
    for (int i = 2; i <= n; i++) {
        if (is_prime[i]) count++;
    }
    return count;
}

int main() {
    // Настройка кириллицы для консоли
    setlocale(LC_ALL, "Russian");

    int num;
    std::cout << "Введите число n для подсчета простых чисел: ";

    // Ручной ввод числа
    if (!(std::cin >> num)) {
        std::cerr << "Ошибка: нужно ввести целое число." << std::endl;
        return 1;
    }

    if (num < 0) {
        std::cout << "Для отрицательных чисел простых чисел нет." << std::endl;
    } else {
        int result = countPrimesNonRecursive(num);
        std::cout << "Количество простых чисел от 2 до " << num << ": " << result << std::endl;
    }

    return 0;
}

```