

**Bài 1. Viết chương trình dùng hàm để đếm xem có bao nhiêu số nguyên tố từ 1 đến n và liệt kê chúng, dùng sàng Eratosthenes.**

**Code:**

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

void SieveOfEratosthenes(int sieve[], int n){
    for (int i = 2; i <= n; i++){
        if (sieve[i]) continue;
        for (int j = 2*i; j <= n; j += i){
            sieve[j] = i;
        }
    }
}

int main(){
    int n, sieve[100000], count = 0;
    cout << "Nhap n (n>1): "; cin >> n;
    for (int i = 0; i < n; i++) sieve[i] = 0;
    SieveOfEratosthenes(sieve,n);
    cout << "Cac so nguyen to tu 1 den " << n << " la: ";
    for (int i = 2; i <= n; i++)
        if (sieve[i] == 0){
            count++;
            cout << i << " ";
        }
    cout << "\nCo tat ca " << count << " so";
}
```

**Độ phức tạp:**  $O(n \cdot \log(n))$

**Màn hình kết quả:**

C:\Users\Admin\Desktop\Bai2.exe

```
Nhap n (n>1): 84
Cac so nguyen to tu 1 den 84 la: 2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83
Co tat ca 23 so
-----
Process exited after 1.182 seconds with return value 0
Press any key to continue . . .
```

## **Bài 2. Viết chương trình thực hiện thuật toán Euclid (Tìm GCD, LCM)**

**Code:**


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

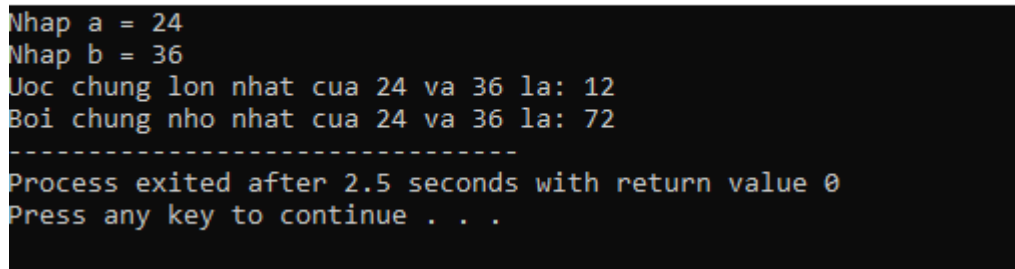
int gcd(int a, int b){
    if (b == 0) return a;
    return gcd(b, a%b);
}

int main(){
    int a, b;
    cout << "Nhap a = "; cin >> a;
    cout << "Nhap b = "; cin >> b;
    cout << "Uoc chung lon nhat cua " << a << " va " << b << " la: " << gcd(a,b);
    cout << "\nBoi chung nho nhat cua " << a << " va " << b << " la: " <<
(a/gcd(a,b))*b;
}
```

**Độ phức tạp:**  $O(\log(\min(a,b)))$

**Màn hình kết quả:**

 C:\Users\Admin\Desktop\Bai2.exe



```
Nhap a = 24
Nhap b = 36
Uoc chung lon nhat cua 24 va 36 la: 12
Boi chung nho nhat cua 24 va 36 la: 72
-----
Process exited after 2.5 seconds with return value 0
Press any key to continue . . .
```

## **Bài 3. Euler's totient function (Tìm số lượng số nguyên tố cùng nhau)**

**Code:**

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

void ptich(int n, int primes[], int exponents[], int &size) {
    size = 0;
    for (int i = 2; i * i <= n; i++) {
        int count = 0;
```

```

        while (n % i == 0) {
            count++;
            n /= i;
        }
        if (count > 0) {
            primes[size] = i;
            exponents[size] = count;
            size++;
        }
    }
    if (n > 1) {
        primes[size] = n;
        exponents[size] = 1;
        size++;
    }
}

int main() {
    int n, primes[100], exponents[100], size;
    cout << "Nhap n (n > 1): "; cin >> n;

    ptich(n, primes, exponents, size);

    long long ans = 1;
    for (int i = 0; i < size; i++) {
        ans *= pow(primes[i], exponents[i] - 1) * (primes[i] - 1);
    }

    cout << "So luong so nguyen to cung nhau voi " << n << " (tu 1 den " << n <<
    ") la: " << ans;
    if (ans == n - 1) cout << " => " << n << " la so nguyen to"; else cout << " => "
    << n << " khong la so nguyen to";

    return 0;
}

```

**Độ phức tạp:**  $O(\sqrt{n})$

**Màn hình kết quả:**

C:\Users\Admin\Desktop\Bai2.exe

```
Nhap n (n > 1): 12
So luong so nguyen to cung nhau voi 12 (tu 1 den 12) la: 4 => 12 khong la so nguyen to
-----
Process exited after 1.234 seconds with return value 0
Press any key to continue . . .
```

**Bài 4.** Cho  $n$  nhập từ bàn phím ( $n$  đủ lớn), phân tích thừa số nguyên tố đối với  $n$ , tìm ước số nguyên tố lớn nhất của nó là  $m$ . Tìm số nguyên tố gần  $n$  nhất. Tìm số hoàn hảo gần  $n$  nhất, liệt kê số hoàn hảo nhỏ hơn  $n$ .

**Code:**

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

```
void ptich(int n, int primes[], int exponents[], int &size) {
    size = 0;
    for (int i = 2; i * i <= n; i++) {
        int count = 0;
        while (n % i == 0) {
            count++;
            n /= i;
        }
        if (count > 0) {
            primes[size] = i;
            exponents[size] = count;
            size++;
        }
    }
    if (n > 1) {
        primes[size] = n;
        exponents[size] = 1;
        size++;
    }
}
```

```
void SieveOfEratosthenes(int sieve[], int n){
    for (int i = 0; i < n; i++) sieve[i] = 0;
    for (int i = 2; i <= n; i++){
        if (sieve[i]) continue;
        for (int j = 2*i; j <= n; j += i){
```

```

        sieve[j] = i;
    }
}

long long tonguoc(int n){
    int primes[100], exponents[100], size;
    ptich(n, primes, exponents, size);

    long long tonguoc = 1;
    for (int i = 0; i < size; i++) {
        int p = primes[i];
        int a = exponents[i];
        long long t = (pow(p, a + 1) - 1) / (p - 1);
        tonguoc *= t;
    }

    return tonguoc;
}

bool shh(int n){
    if (n == tonguoc(n) - n) return true; else return false;
}

int main() {
    int n, primes[100], exponents[100], size, sieve[1000000];
    cout << "Nhap n (n > 1): "; cin >> n;

    ptich(n, primes, exponents, size);

    cout << "Phan tich thua so nguyen to cua " << n << ", ta duoc: ";
    for (int i = 0; i < size; i++) {
        cout << primes[i] << "^" << exponents[i];
        if (i != size - 1) {
            cout << " * ";
        }
    }

    SieveOfEratosthenes(sieve, 1000000);
    cout << "\nUoc nguyen to lon nhat cua " << n << " la: " << sieve[n];
}

```

```
cout << "\nSo nguyen to gan " << n << " nhat la: ";
```

```
if (sieve[n] == 0) cout << n << endl; else {  
    int ans1 = -1, ans2 = -1;  
    for (int i = n - 1; i >= 2; i--)  
        if (sieve[i] == 0) {  
            ans1 = i;  
            break;  
        }  
    for (int i = n + 1; i <= 100000; i++)  
        if (sieve[i] == 0) {  
            ans2 = i;  
            break;  
        }  
    if (n - ans1 < ans2 - n) cout << ans1 << endl; else cout << ans2 << endl;  
}
```

```
cout << "Liet ke cac so hoan hao nho hon " << n << ": ";  
int ans1 = -1;  
for (int i = 6; i <= n; i++)  
    if (shh(i)){  
        cout << i << " ";  
        ans1 = i;  
    }
```

```
cout << "\nSo hoan hao gan " << n << " nhat la: ";
```

```
if (shh(n)) cout << n << endl; else {  
    int ans2 = -1;  
    for (int i = n + 1; i <= 100000; i++)  
        if (shh(i)) {  
            ans2 = i;  
            break;  
        }  
    if (ans2 == -1) cout << ans1; else  
    if (n - ans1 < ans2 - n) cout << ans1 << endl; else cout << ans2 << endl;  
}  
return 0;  
}
```

**Độ phức tạp:**  $O(n \cdot \sqrt{n} \cdot \log(n))$

**Màn hình kết quả:**

```
C:\Users\Admin\Desktop\Bai2.exe
Nhap n (n > 1): 9876
Phan tich thua so nguyen to cua 9876, ta duoc: 2^2 * 3^1 * 823^1
Uoc nguyen to lon nhat cua 9876 la: 823
So nguyen to gan 9876 nhat la: 9871
Liet ke cac so hoan hao nho hon 9876: 6 28 496 8128
So hoan hao gan 9876 nhat la: 8128
-----
Process exited after 1.918 seconds with return value 0
Press any key to continue . . .
```