Code:

|  |
| --- |
| #include <iostream>  #include <vector>  using namespace std;  const int N = 1e6+5;  bool E[N+5];  bool isPrime(long long n){      if (n <= 1) return false;      if (n==2||n==3) return true;      if (n%2==0||n%3==0) return false;      for (int i=5;i\*i<=n;i+=6){          if (n%i==0||n%(i+2)==0) return false;      }      return true;  }  void Eratos(long long n){ // O(N\*logN)      for (int i=1;i<=n;++i)  E[i] = true;      E[0] = E[1] = false;      for (int i=2;i\*i<=n;++i){          if (E[i] == true){              for (int j=i\*i;j<=n;j+=i){                  E[j] = false;              }          }      }  }  void Bai1(long long n){      Eratos(n);      vector<int>vt;      for (int i=2;i<=n;++i){          if (E[i] == true)   vt.push\_back(i);      }      cout << "So luong so nguyen to: " << vt.size() << '\n';      cout << "Danh sach so nguyen to: \n";      for (int x : vt)    cout << x << ' ';      cout << '\n';  }  long long euclid\_gcd(long long a,long long b){ // O(logN)      if (b==0)   return a;      return euclid\_gcd(b,a%b);  }  int lcm(long long a,long long b){      return a/euclid\_gcd(a,b)\*b;  }  void Bai2(long long a,long long b){      cout << "GCD cua " << a << " va " << b << " la: " << euclid\_gcd(a,b) << '\n';      cout << "LCM cua " << a << " va " << b << " la: " << lcm(a,b) << '\n';  }  long long Euler\_totien\_function(long long n){ // O(sqrt(N))      if (n==0)   return 0;      long long ans = n;      for (int i=2;i\*i<=n;++i){          if (n%i==0){              ans -= ans/i;              while (n%i==0)  n/=i;          }      }      if (n>1)    ans -= ans/n;      return ans;  }  void Print\_Coprimes(long long n){      for (int i=1;i<=n;++i){          if (euclid\_gcd(i,n)==1) cout << i << ' ';      }  }  void Bai3(long long n){      cout << "Ket qua cua ham Phi Euler cho n = " << n << " la: ";      cout << Euler\_totien\_function(n) << '\n';      cout << "Danh sach so nguyen to cung nhau voi " << n << ": ";      Print\_Coprimes(n);      cout << '\n';  }  void Bai4(long long n){      long long left=n-1;      long long right=n+1;      cout << "So nguyen to gan " << n << " nhat la: ";      while(true){          if (isPrime(left)){              cout << left;              break;          }          if (isPrime(right)){              cout << right;              break;          }          right++;          left--;      }      cout << '\n';  }  void factorize(long long n,vector<int>& p,vector<int>& a){      if (n<=1){          cout << "n > 1 moi co phan tich thua so nguyen to\n";          return;      }      p.clear();      a.clear();      for (int i=2;i\*i<=n;++i){          if (n%i==0){              p.push\_back(i);              int cnt=0;              while (n%i==0){                  cnt++;                  n/=i;              }              a.push\_back(cnt);          }      }      if (n>1){          p.push\_back(n);          a.push\_back(1);      }  }  bool isPerfect(long long n){      if (n<=1) return false;      long long sum=1;      for (int i=2;i\*i<=n;++i){          if(n%i==0){              if (n/i==i) sum+=i;              else sum += (i+n/i);          }      }      return sum==n;  }  void Bai5(long long n){      vector<int>p,a;      factorize(n,p,a);      if(!p.empty()){          cout << "Uoc so nguyen to lon nhat cua " << n << " la: " << p.back() << '\n';      }      long long left=n-1;      long long right=n+1;      cout << "So hoan hao gan " << n << " nhat la: ";      while(true){          if (left>0&&isPerfect(left)){              cout << left;              break;          }          else left--;          if (isPerfect(right)){              cout << right;              break;          }          else right++;      }      cout << '\n';  }  int main()  {      long long n,a,b;      cout << "Nhap n cho bai 1: ";      cin >> n;      Bai1(n);      cout << "Nhap a va b cho bai 2:";      cin >> a >> b;      Bai2(a,b);      cout << "Nhap n cho bai 3: ";      cin >> n;      Bai3(n);      cout << "Nhap n cho bai 4: ";      cin >> n;      Bai4(n);      cout << "Nhap n cho bai 5: ";      cin >> n;      Bai5(n);      return 0;  } |

Kết quả:

A screenshot of a computer

Description automatically generated

Phân tích độ phức tạp của từng thuật toán.

Hàm isPrime – Kiểm tra tính nguyên tố của một số nguyên dương. Độ phức tạp thời gian O, độ phức tạp bộ nhớ O(1)

Hàm Eratos – Sàng các số nguyên tố từ 1->N. Độ phức tạp thời gian O, độ phức tạp bộ nhớ O(N)

Hàm euclid\_gcd – Tính ước chung lớn nhất của 2 số nguyên. Độ phức tạp thời gian O(logN), độ phức tạp bộ nhớ O(logN) (Do cài đặt bằng đệ quy)

Hàm Euler\_totien\_function – Tính số lượng số từ 1 -> N nguyên tố cùng nhau với N. Độ phức tạp thời gian O, độ phức tạp bộ nhớ O(1)