**Lý Thuyết Số**

1. Bài tập trên lớp
   1. Cài đặt thuật toán Eratosthenes. (sử dụng ngôn ngữ tùy ý C/C++, Java…) Inp: Số nguyên dương N

Out: Dãy có các số nguyên tố từ 1->N

* 1. Cài đặt thuật toán Euclidean mở rộng để tìm bộ số (d, x, y) cho biểu thức sau: ax + by = d

Trong đó: a, b là các số nguyên không âm d = gcd(a, b)

Input: 2 số nguyên a, b Output: (x, y)

**Bài giải:**

Bài 1:

#include <iostream>

using namespace std;

int main() {

int N;

cout<<"Nhap N:";

cin>>N;

bool check[N + 1];

// Khởi tạo tất cả các số [2...N] đều là số nguyên tố

for (int i = 2; i <= N; i++) {

check[i] = true;

}

// Thuật toán sàng nguyên tố

// Nếu một số là số nguyên tố, thì tất cả các bội của nó không phải số nguyên tố

for (int i = 2; i <= N; i++) {

if (check[i] == true) {

for (int j = 2 \* i; j <= N; j += i) {

check[j] = false;

}

}

}

// In ra các số là số nguyên tố

for (int i = 2; i <= N; i++) {

if (check[i] == true) {

printf("%d ", i);

}

}

}

Bài 2:

#include <iostream>

using namespace std;

// Hàm tính gcd(a, b) bằng thuật toán Euclidean

int gcd(int a, int b, int &x, int &y) {

if (b == 0) {

x = 1;

y = 0;

return a;

}

int x1, y1;

int d = gcd(b, a % b, x1, y1);

x = y1;

y = x1 - (a / b) \* y1;

return d;

}

int main() {

int a, b;

cout << "Nhap a: ";

cin >> a;

cout << "Nhap b: ";

cin >> b;

int x, y, d;

d = gcd(a, b, x, y);

cout << "GCD(" << a << ", " << b << ") = " << d << endl;

cout << "x = " << x << ", y = " << y << endl;

return 0;

}