**KỸ THUẬT LẬP TRÌNH**

**Bài tập Lab 3**

**---o0o---**

**PHẦN 1 - HÀM**

**Bài 1**. Viết hàm tìm số đảo ngược của 1 số nguyên

Ví dụ: - input: 234543

- output: 345432

#include<iostream>

**int** soDaoNguoc(**int** n) {

**int** donvi = n % 10;

**int** chuc = n / 10 % 10;

**int** tram = n / 100 % 10;

**int** nghin = n / 1000 % 10;

**int** trucnghin = n / 10000 % 10;

**int** tramnghin = n / 100000 % 10;

**int** tong = (donvi \* 100000) + (chuc \* 10000) + (tram \* 1000) + (nghin \* 100) + (trucnghin \* 10) + (tramnghin) ;

    std::cout << tong;

}

**int** main() {

**int** n;

    std::cout << "Nhap mot day so ban muon dao nguoc:  ";

    std::cin >> n;

    soDaoNguoc(n);

}

**Bài 2**. Viết hàm trả về chiều dài của một số nguyên.

Ví dụ: - input: 547897

- output: 6

#include <iostream>

**void** Demchieudai(**int** n){

**int** dem = 0;

    while (n != 0)

    {

        n /= 10;

        ++dem;

    }

    std::cout << dem;

}

**int** main(){

**long** n;

    std::cout << "Nhap so ban muon dem do dai: ";

    std::cin >> n;

    Demchieudai(n);

}

**Bài 3**. Viết hàm tìm phần tử thứ n của dãy fibonacci.

#include <iostream>

**int** demson(**int** n)

{

*// vì so Fibonacci rất lớn nên để long long*

**long** **long** fn, f0 = 0, f1 = 1;

    if (n < 0){

        std::cout << "Ban da nhap sai so!" << "\n";

    }

    else {

        if (n == 0){

            std::cout << "0" << "\n";

        }

        if (n == 1){

            std::cout << "1" << "\n";

        }

        if (n >= 2){

            for (**int** i = 2; i <= n; i++)

            {

*// f0 <=> f (n - 1);*

*// f1 <=> f (n - 2)*

                fn = f0 + f1;

                f0 = f1;

                f1 = fn;

            }

            std::cout << "So = " << fn << "\n";

            std::cout << "So 1 = " << f0 << "\n";

            std::cout << "So 2 = " << f1 << "\n";

        }

    }

}

**int** main()

{

**int** n;

    std::cout << "Nhap so Fibonacci = ";

    std::cin >> n;

    demson(n);

}

**Bài 4**. Viết hàm chuyển đổi số hệ thập phân sang số hệ nhị phân hoặc thập lục phân. *(Hàm sẽ có tham số để xác định đầu ra là số hệ 2 hay hệ 16).*

#include <iostream>

**void** InraNhiPhan(**int** n){

    if (n == 0) { *// nếu số chia = 0 thì chúng ta dừng lại*

        return;

    }

    else{

**int** r = n % 2; *// chia lay du vd: 2 % 2 = 1 du 0 no se lay 0;*

        InraNhiPhan(n/2); *// vd 2/2 duoc 1 lay nguyen 1*

        std::cout << r;

    }

}

**int** main(){

**int** n;

    do

    {

        std::cout << "Nhap vao so muon doi ra Nhi phan = ";

        std::cin >> n;

    } while (n < 0);

    InraNhiPhan(n);

}

**Bài 5**. Viết hàm tính tổ hợp chập k của n phần tử nếu tồn tại.

#include <iostream>

**int** ToHop(**int** k, **int** n)

{

    if (k == n || k == 0)

    {

        return 1;

    }

    return ToHop(k, n - 1) + ToHop(k - 1, n - 1);

}

**int** main()

{

**int** k, n;

**char** Tieptuc;

    do

    {

        std::cout << "Moi ban nhap k = ";

        std::cin >> k;

        std::cout << "Moi ban nhap n = ";

        std::cin >> n;

        if (k >= n)

        {

            do

            {

                std::cout << "Moi ban nhap lai k, k < n"

                          << "\n";

                std::cout << "Moi ban nhap k = ";

                std::cin >> k;

                std::cout << "Moi ban nhap n = ";

                std::cin >> n;

            } while (k >= n);

        }

        std::cout << "\n To hop C(" << k << "," << n << ") = " << ToHop(k, n) << "\n";

        std::cout << "Ban co muon tiep tuc khong: Y/n: ";

        std::cin >> Tieptuc;

    } while (Tieptuc == 'y' || Tieptuc == 'Y');

    return 0;

}

**Bài 6**: Viết chương trình nhập vào 2 số nguyên M, N (M<N). Tính tổng các số

chẵn trong dãy số Fibonaccy từ F(M) đến F(N).

#include<iostream>

**int** chanletong(**int** m, **int** n){

**int** fn , f0 = 0, f1 = 1;

**int** fm , f2 = 0, f3 = 1;

**int** tong\_1 = 0;

**int** tong\_2 = 0;

**int** tongcacsochan = 0;

    if (n < 0 ||  m < 0){

        std::cout << "Ban da nhap sai so! \n";

    }

    else {

        if (n == 0){

            std::cout << "n = 0 \n";

            if (m == 0){

                std::cout << "m = 0 \n";

            }

        }

            if (n == 1){

                std::cout << "n = 1 \n";

                if ( m == 1){

                    std::cout << "m = 1 \n";

                }

            }

        }

        if (n >= 2){

            for (**int** i = 2; i <= n; i++)

            {

               fn = f0 + f1;

               f0 = f1;

               f1 = fn;

               if (fn % 2 == 0){

                tong\_1 = tong\_1  + fn;

                    std::cout << "Fn chan = " << tong\_1 << "\n";

               }

            }

            std::cout <<"Fn = " << fn << "\n";

            if (m >= 2){

                for (**int** j = 2; j <= n; j++)

                {

                    fm = f2 + f3;

                    f2 = f3;

                    f3 = fm;

                    if (fm % 2 == 0)

                    {

                        tong\_2 = tong\_2 + fm;

                        std::cout << "Fm chan = " << tong\_2 << "\n";

                    }

                }

                std::cout << "Fm = "<< fm << "\n";

            }

        }

        tongcacsochan = tong\_1 + tong\_2;

        std::cout << "Tong so chan trong Fibonacci = " << tongcacsochan << "\n";

}

**int** main(){

**int** n,m;

    std::cout << "Nhap so Fibonacci thu 1: ";

    std::cin >> n;

    std::cout << "Nhap vao so Fibonacci thu 2: ";

    std::cin >> m;

    chanletong(m, n);

}

**Bài 7:** Viết chương trình nhập vào 3 số nguyên a, b, c.

Tính: S=USCLN(a,b)+USCLN(b,c)+USCLN(a,c)

#include <iostream>

**int** USCLNa(**int** a, **int** b){

    while (a != b)

    {

        if (a > b)

        {

            a = a - b;

        }

        else

        {

            b = b - a;

        }

    }

    return a;

}

**int** USCLNb(**int** b, **int** c){

    while (b != c)

    {

        if (b > c)

        {

            b = b - c;

        }

        else

        {

            c = c - b;

        }

    }

    return b;

};

**int** USCLNc (**int** c , **int** a){

    while (a != c)

    {

        if (a > c)

        {

            a = a - c;

        }

        else

        {

            c = c - a;

        }

    }

    return c;

}

**int** USCLN(**int** a, **int** b, **int** c){

**int** max = 0;

    std::cout << "Uoc (" <<a<< ")" << " = ";

    for (**int** i = 1; i <= a; i++)

    {

        if (a % i == 0){

            std::cout << i << ",";

        }

    }

    std::cout << "\n";

    std::cout << "Uoc (" << b << ")" << " = ";

    for (**int** j = 1; j <= b; j++)

    {

        if (b % j == 0)

        {

            std::cout << j << ",";

        }

    }

    std::cout <<"\n";

    std::cout << "Uoc (" << c << ")" << " = ";

    for (**int** z = 1; z <= b; z++)

    {

        if (c % z == 0)

        {

            std::cout << z << ",";

        }

    }

**int** test = 0;

}

**int** main(){

**int** a, b, c;

    std::cout << "Nhap a = ";

    std::cin >> a;

    std::cout << "Nhap b = ";

    std::cin >> b;

    std::cout << "Nhap c = ";

    std::cin >> c;

    USCLN(a, b, c);

    std::cout << "\n";

    std::cout << "UCLN(a, b) + UCLN(b, c) + UCLN(a, c) = " << USCLNa(a, b) + USCLNb(b, c) + USCLNc(a, c);

}

**Bài 9:** Viết hàm tính giá trị đa thức bậc n: *F*( )*x* = *a xn n* + *an*−1*xn*−1 +....+ *a x*1 + *a*0 , hàm có 3 tham số là mảng số thực các hệ số ai, giá trị n, x. (*Không* đượ*c s*ử *d*ụ*ng hàm l*ũ*y th*ừ*a s*ẵ*n có*)

#include <iostream>

#include<cmath>

**int** main(){

    std::cout << "Chuong trinh tinh gia tri da thuc bac n! \n";

**int** n ;

    std::cout << "Nhap cap bac cua da thuc n = ";

    std::cin >> n;

**float** a[100];

    for (**int** i = n; i >= 0; --i)

    {

        std::cout << "Nhap he so cua x^" << i << " = ";

        std::cin >> a[i];

    }

    std::cout << "Da thuc vua nhap f(x) = ";

    for (**int** i = n; i >= 0; --i)

    {

        if (i == 0){

            std::cout << a[i] ;

        }

        else{

            std::cout << a[i] << "x^" << i << " + ";

        }

    }

    std::cout << "\n";

**long** **long** fx = 0;

**int** x;

    std::cout << "Nhap gia tri cua x = ";

    std::cin >> x;

    for (**int** i = n; i >= 0; --i)

    {

        fx = fx + a[i] \* pow(x, i);

    }

    std::cout << "f( " << x << " ) = " << fx ;

}

**PHẦN 2 – MẢNG**

**Bài 1**. Nhập/xuất dữ liệu cho mảng số nguyên có tối đa 100 phần tử.

a. Kiểm tra xem mảng có phải chứa toàn số chẵn hay không?

#include <iostream>

#include <cmath>

**int** Nhaparr(**int** a[],**int** n){

    for (**int** i = 0; i < n; i++)

    {

        std::cout << "a [" << i << "] = " ;

        std::cin >> a[i];

    }

}

**int** Xuatarr (**int** a[], **int** n){

    std::cout << "Ta co mang => " ;

    for (**int** i = 0; i < n; i++)

    {

        std::cout << a[i] << ", ";

    }

}

*// 1.a  Kiểm tra xem mảng có phải chứa toàn số chẵn hay không?*

**bool** chanle (**int** a[], **int** n){

    for (**int** i = 0; i < n; i++)

    {

        if (a[i] % 2 == 0)

        {

            return true;

        }

        return false;

    }

}

**int** main(){

**int** \*a, n;

    a = new **int**(n);

    std::cout << "Nhap do dai mang = ";

    std::cin >> n;

    Nhaparr(a, n);

    Xuatarr(a, n);

    std::cout << "\n";

    if (chanle(a, n) == true){

        std::cout << "Mang toan so chan";

    }

   else {

        std::cout << "Mang kh co so chan";

    }

}

b. Hiển thị các số nguyên tố có trong mảng ?

*// // b. Hiển thị các số nguyên tố có trong mảng ?*

#include <iostream>

#include <cmath>

**int** Nhaparr(**int** a[],**int** n){

    for (**int** i = 0; i < n; i++)

    {

        std::cout << "a [" << i << "] = " ;

        std::cin >> a[i];

    }

}

**int** Xuatarr (**int** a[], **int** n){

    std::cout << "Ta co mang => " ;

    for (**int** i = 0; i < n; i++)

    {

        std::cout << a[i] << ", ";

    }

}

**bool** kiemtrasonguyento(**int** n){

    if (n < 2){

        return false;

    }

    else {

        if (n == 2){ *// so 2 la so chan duy nhat la so nguyen to*

            return true;

        }

        else {

            if (n % 2 == 0){

                return false;

            }

            else {

                for (**int** i = 2; i < n; i++)

                {

                    if (n % i == 0){

                        return false;

                    }

                }

            }

        }

    }

    return true;

}

**void** coutsonguyento(**int** a[], **int** n){

*// duyet tu dau mang den cuoi mang lay tung phan tu trong mang bo vao ham kiem tra so nguyen to;*

    std::cout << " => So nguyen to trong mang = ";

    for (**int** i = 0; i < n; i++)

    {

        if (kiemtrasonguyento(a[i]) == true)

        {

            std::cout << a[i] << "; ";

        }

    }

    std::cout << "\n";

    std::cout << " => Khong phai la so nguyen to trong mang = ";

    for (**int** i = 0; i < n; i++)

    {

        if (kiemtrasonguyento(a[i]) == false)

        {

            std::cout << a[i] << "; ";

        }

    }

}

*// . Mảng có đang được sắp xếp hay không? Nếu có thì mảng đã sắp tăng dần hay giảm dần?*

**bool** mangtangdan (**int** n, **int** a[]){

**int** num = 1;

    for (**int** i = 0; i <  n - 1; i++)

    {

        if (a[i] > a[i + 1])

        {

            num = 0;

            break;

        }

    }

    return num;

}

**int** main(){

**int** \*a, n;

    a = new **int**(n);

    std::cout << "Nhap do dai mang = ";

    std::cin >> n;

    Nhaparr(a, n);

    Xuatarr(a, n);

coutsonguyento(a, n);

}

c. Mảng có đang được sắp xếp hay không? Nếu có thì mảng đã sắp tăng dần hay giảm dần?

#include <iostream>

#include <cmath>

**int** Nhaparr(**int** a[],**int** n){

    for (**int** i = 0; i < n; i++)

    {

        std::cout << "a [" << i << "] = " ;

        std::cin >> a[i];

    }

}

**int** Xuatarr (**int** a[], **int** n){

    std::cout << "Ta co mang => " ;

    for (**int** i = 0; i < n; i++)

    {

        std::cout << a[i] << ", ";

    }

}

**bool** mangtangdan (**int** n, **int** a[]){

**int** num = 1;

    for (**int** i = 0; i <  n - 1; i++)

    {

        if (a[i] > a[i + 1])

        {

            num = 0;

            break;

        }

    }

    return num;

}

**bool** manggiamdan (**int** n, **int** a[]){

**int** num = 1;

    for (**int** i = 0; i <  n - 1; i++)

    {

        if (a[i] < a[i + 1])

        {

            num = 0;

            break;

        }

    }

    return num;

}

**int** main(){

**int** \*a, n;

    a = new **int**(n);

    std::cout << "Nhap do dai mang = ";

    std::cin >> n;

    Nhaparr(a, n);

    Xuatarr(a, n);

**int** tang = mangtangdan(n, a);

**int** giam = manggiamdan(n ,a);

if (tang == 1){

        std::cout << " => Mang sap tang dan";

    }

    else{

        std::cout << " => Mang sap giam dan";

    }

    std::cout << "\n";

}

d. Cho biết mảng có các phần tử trùng nhau hay không? Nếu có, hãy loại bỏ bớt và chỉ để lại 1 phần tử.

*// d. Cho biết mảng có các phần tử trùng nhau hay không? Nếu có, hãy loại bỏ bớt và chỉ để lại 1 phần tử.*

*// Hàm xử lý xóa các phần tử trùng nhau tại vị trí bất kì;*

#include <iostream>

#include <cmath>

**int** Nhaparr(**int** a[],**int** n){

    for (**int** i = 0; i < n; i++)

    {

        std::cout << "a [" << i << "] = " ;

        std::cin >> a[i];

    }

}

**int** Xuatarr (**int** a[], **int** n){

    std::cout << "Ta co mang => " ;

    for (**int** i = 0; i < n; i++)

    {

        std::cout << a[i] << ", ";

    }

}

*// + &n là sau khi xóa 1 giá trị trong mảng ta sẽ lưu giá trị đó lại trong n tham chiếu*

**void** XoaBatKy(**int** a[], **int** **&**n, **int** vitri)

{

    for (**int** i = vitri + 1; i < n; i++)

    {

        a[i - 1] = a[i]; *// ví dụ a[3] sau khi xóa nó sẽ về vị trí a[2] để đứng*

    }

    n --; *// sau khi xóa xong a[i - 1] nó sẽ giảm nếu trừ vì vậy ta nên giảm độ dài mảng*

}

**void** xoa\_phan\_tu\_trung\_nhau(**int** a[], **int** **&**n){ *// xóa phần tử sl tham chiếu sẽ giảm đi có tham chiếu lưu n để nso giảm đi*

    for (**int** i = 0; i < n - 1; i++) *// n - 1 phần tử cuối cùng nó sẽ là số phân biệt kh duyet nữa*

    {

        for (**int** j = i + 1; j < n ; j++)

        {

            if (a[i] == a[j]){

               XoaBatKy(a, n, j) ;*// cần xóa vị trí thứ a[j] -> j*

               i--;

            }

        }

    }

}

**int** main(){

**int** \*a, n;

    a = new **int**(n);

    std::cout << "Nhap do dai mang = ";

    std::cin >> n;

    Nhaparr(a, n);

    Xuatarr(a, n);

 xoa\_phan\_tu\_trung\_nhau(a, n);

    std::cout << "Mang sau khi xoa phan tu trung nhau => ";

    Xuatarr(a, n);

}

e. Hãy so sánh số lượng số chẵn so với số lượng số lẻ có trong mảng?

#include <iostream>

#include <cmath>

**int** Nhaparr(**int** a[],**int** n){

    for (**int** i = 0; i < n; i++)

    {

        std::cout << "a [" << i << "] = " ;

        std::cin >> a[i];

    }

}

**int** Xuatarr (**int** a[], **int** n){

    std::cout << "Ta co mang => " ;

    for (**int** i = 0; i < n; i++)

    {

        std::cout << a[i] << ", ";

    }

}

**void** demslchanle(**int** a[], **int** n){

**int** demchan = 0;

**int** demle = 0;

    for (**int** i = 0; i < n; i++)

    {

        if (a[i] % 2 == 0){

            demchan ++;

        }

    }

    for (**int** i = 0; i < n; i++)

    {

        if (a[i] % 2 != 0)

        {

            demle ++;

        }

    }

    std::cout << "So luong chan trong mang = "  << demchan << "\n";

    std::cout << "So luong le trong mang = "  << demle;

    std::cout << "\n";

    if (demchan > demle)

    {

        std::cout << "Chan lon hon le";

    }

    else if (demle > demchan){

        std::cout << "le lon hon chan";

    }

    else {

        std::cout << "Chan bang le";

    }

}

**int** main(){

**int** \*a, n;

    a = new **int**(n);

    std::cout << "Nhap do dai mang = ";

    std::cin >> n;

    Nhaparr(a, n);

    Xuatarr(a, n);

demslchanle(a, n);

**}**

f. Hãy chèn số có giá trị x vào vị trí có chỉ số p của mảng nếu mảng chưa đầy?

*// f.Hãy chèn số có giá trị x vào vị trí có chỉ số p của mảng nếu mảng chưa đầy?*

#include <iostream>

#include <cmath>

**int** Nhaparr(**int** a[],**int** n){

    for (**int** i = 0; i < n; i++)

    {

        std::cout << "a [" << i << "] = " ;

        std::cin >> a[i];

    }

}

**int** Xuatarr (**int** a[], **int** n){

    std::cout << "Ta co mang => " ;

    for (**int** i = 0; i < n; i++)

    {

        std::cout << a[i] << ", ";

    }

}

**void** themphantuvaovitribatki(**int** a[], **int** **&**n, **int** x, **int** vitri){

*// vòng lặp này làm công việc để dịch từng phầng tử từ thàng n - 1 về đến vitri*

    for (**int** i = n - 1; i >= vitri; i--) *// n - 1 là thằng cuối cùng dịch đi trước*

*// i-- nó sẽ dịch xét từ n-1 tới về sau*

    {

        a[i + 1] = a[i]; *// a[i] = 4 thì a[i + 1] sẽ là 5 chúng ta sẽ gán từ phải qua trái tức là a[i] = a[i + 1] ;*

*// sau khi thêm nó sẽ nằm ở a[i];*

    }

    a[vitri] = x; *// thêm vị trí x vào bất kì trong mảng*

    n ++; *// sau khi thêm nó sẽ tăng đọ dài*

}

int main(){

**int** \*a, n;

    a = new **int**(n);

    std::cout << "Nhap do dai mang = ";

    std::cin >> n;

    Nhaparr(a, n);

    Xuatarr(a, n);

**int** x;

    std::cout << "Nhap phan tu can them = ";

    std::cin >> x;

**int** vitri;

    do

    {

        std::cout << "Nhap vi tri can them = ";

        std::cin >> vitri;

        if (vitri < 0 || vitri > n){

            std::cout << "\n vi tri can them trong doan tu [0, " << n << "]";

        }

    } while (vitri < 0 || vitri > n);

    themphantuvaovitribatki(a, n, x, vitri);

    Xuatarr(a, n);

}

g. Hãy tìm số nguyên tố đầu tiên xuất hiện trong mảng mà phần tử đứng trước của nó là số chính phương?

*// g. Hãy tìm số nguyên tố đầu tiên xuất hiện trong mảng mà phần tử đứng trước của nó là số chính phương?*

#include <iostream>

#include <cmath>

**int** Nhaparr(**int** a[],**int** n){

    for (**int** i = 0; i < n; i++)

    {

        std::cout << "a [" << i << "] = " ;

        std::cin >> a[i];

    }

}

**int** Xuatarr (**int** a[], **int** n){

    std::cout << "Ta co mang => " ;

    for (**int** i = 0; i < n; i++)

    {

        std::cout << a[i] << ", ";

    }

}

**bool** kiem\_tra\_so\_chinh\_phuong(**int** n){

**int** ketqua = (**int**)sqrt(n \* 1.0);

    if (ketqua \* ketqua == n) *// nhân lại ra bình phương*

    {

        return true; *// là số chính phương*

    }

    return false; *// ko phải là số chính phương*

}

**void** so\_chinh\_phuong(**int** a[], **int** n)

{

*// số chính phương khi căng ra thì nó sẽ là số nguyên*

    std::cout << "So chinh phuong la = ";

    for (**int** i = 0; i <= n; i++)

    {

        if (kiem\_tra\_so\_chinh\_phuong(a[i]) == true)

        {

            std::cout << a[i] << " ";

        }

        else

        {

            std::cout << "Mang kh co so chinh phuong";

        }

    }

    std::cout << "\n";

}

**int** main(){

**int** \*a, n;

    a = new **int**(n);

    std::cout << "Nhap do dai mang = ";

    std::cin >> n;

    Nhaparr(a, n);

    Xuatarr(a, n);

 so\_chinh\_phuong(a, n);

    std::cout << "Mang cu = ";

    Xuatarr(a, n);

}

h. Tách các số không phải là số nguyên tố và đưa vào mảng khác.

#include <iostream>

#include <cmath>

**int** Nhaparr(**int** a[],**int** n){

    for (**int** i = 0; i < n; i++)

    {

        std::cout << "a [" << i << "] = " ;

        std::cin >> a[i];

    }

}

**int** Xuatarr (**int** a[], **int** n){

    std::cout << "Ta co mang => " ;

    for (**int** i = 0; i < n; i++)

    {

        std::cout << a[i] << ", ";

    }

}

*// h. Tách các số không phải là số nguyên tố và đưa vào mảng khác.*

**bool** kiem\_tr\_sn(**int** n){ *// n = a[i]*

    if (n < 2)

    {

        return false;

    }

    else

    {

        if (n == 2)

        {

            return false;

        }

        else

        {

            for (**int** i = 2; i < n; i++)

            {

                if (n % i == 0)

                {

                    return false;

                }

            }

        }

    }

    return true;

}

**void** in\_ra\_pt\_la\_snt(**int** a[], **int** **&**n)

{

    std::cout << "\n\n----Mang chua so nguyen to----\n";

    std::cout <<  "Mang new = ";

    for (**int** i = 0; i <= n; i++)

    {

        if (kiem\_tr\_sn(a[i]) == true){

            std::cout << a[i] << "; ";

        }

    }

    std::cout << "\n\n----Mang chua so khong phai so nguyen to----\n";

    std::cout << "Mang new = ";

    for (**int** i = 0; i <= n; i++)

    {

        if (kiem\_tr\_sn(a[i]) == false)

       {

            std::cout << a[i] << "; ";

       }

    }

}

**int** main(){

**int** \*a, n;

    a = new **int**(n);

    std::cout << "Nhap do dai mang = ";

    std::cin >> n;

    Nhaparr(a, n);

    Xuatarr(a, n);

in\_ra\_pt\_la\_snt(a, n);

}

i. Sắp xếp nửa đầu của mảng tăng dần, nửa sau giảm dần. Phần tử ở giữa nếu có thì giữ nguyên.

#include <iostream>

using **namespace** std;

*// cau i*

*// ham hoan doi gia tri 2 bien k l*

**void** hoandoi(**int** **&**k, **int** **&**l)

{

**int** bientam = k; *// bien tam de luu k*

    k = l;

    l = bientam;

}

*// ham sap xep nua dau tang nua sau giam*

**void** sapxepnuatangnuagiam(**int** a[], **int** n)

{

    if (n <= 1) *// neu so luong so trong mang nho hon hoac bang 1 thi dung lai*

    {

        return;

    }

*// vi mang bat dau tu 0 nen khi nhap n chan thi tong so luong cac chu so trong mang tu 0 den n se la le va nguoc lai*

    if (n % 2 != 0) *// neu n la so le thi ko can tim so chinh giua*

    {

*// sap xep nua dau tang dan*

        for (**int** i = 0; i <= n / 2; i++) *// tao vong lap i tu 0 den n / 2*

        {

            for (**int** j = i + 1; j <= n / 2; j++) *// tao vong lap j tu i + 1 den n / 2*

            {

                if (a[i] > a[j]) *// neu a[i] > a[j] thi hoandoi*

                {

                    hoandoi(a[i], a[j]);

                }

            }

            cout << a[i] << " "; *// in ra dáy nua dau da sap xep tang dan*

        }

    }

    else *// neu n la so chan thi tim so chinh giua*

    {

*// sap xep nua dau tang dan*

        for (**int** i = 0; i < n / 2; i++) *// tao vong lap i tu 0 den n / 2*

        {

            for (**int** j = i + 1; j < n / 2; j++) *// tao vong lap j tu i + 1 den n / 2*

            {

                if (a[i] > a[j]) *// neu a[i] > a[j] thi hoandoi*

                {

                    hoandoi(a[i], a[j]);

                }

            }

            cout << a[i] << " "; *// in ra day nua dau da xep tang dan*

        }

**int** sogiua = a[n / 2]; *// tim so o giua va in ra neu nhap n la so chan*

        cout << sogiua << " ";

    }

*// sap xep nua sau giam dan*

    for (**int** i = n / 2 + 1 ; i <= n; i++) *// tao vong lap i tu n / 2 + 1 den n*

    {

        for (**int** j = i + 1; j <= n; j++) *// tao vong lap j tu i + 1 den n*

        {

            if (a[i] < a[j]) *// neu a[i] < a[j] thi hoandoi*

            {

                hoandoi(a[i], a[j]);

            }

        }

        cout << a[i] << " "; *// in ra day nua sau da sap xep giam dan*

    }

}

**int** main()

{

**int** n;

    cout << "nhap do dai mang: ";

    cin >> n;

**int** \*a; *// con tro*

    a = new **int**(n);

*// int mang = a[];*

**int** num;

    for (**int** i = 0; i <= n; i++) *// tao vong lap de nhap mang*

    {

        cout << "a[" << i << "]";

        cin >> a[i];

    }

    cout << "---mang da nhap---" << endl;

    for (**int** i = 0; i <= n; i++) *// tao vong lap de xuat mang*

    {

        cout << "a[" << i << "] = " << a[i] << "\n";

    }

*// cau i*

    cout << "sap xep nua dau tang nua sau giam: ";

    sapxepnuatangnuagiam(a, n);

}

k. Sắp xếp mảng giảm dần.

#include <iostream>

#include <cmath>

**int** Nhaparr(**int** a[],**int** n){

    for (**int** i = 0; i < n; i++)

    {

        std::cout << "a [" << i << "] = " ;

        std::cin >> a[i];

    }

}

**int** Xuatarr (**int** a[], **int** n){

    std::cout << "Ta co mang => " ;

    for (**int** i = 0; i < n; i++)

    {

        std::cout << a[i] << ", ";

    }

}

*// k. sắp xép mảng giảm dần*

**void** mang\_giam\_dan\_di(**int** a[], **int** n){

**int** bien\_chua\_tam;

    for (**int** i = 0; i < n; i++)

    {

        for (**int** j = i + 1; j < n; j++)

        {

            if (a[i] < a[j])

            {

                bien\_chua\_tam = a[i];

                a[i] = a[j];

                a[j] = bien\_chua\_tam;

            }

        }

    }

}

**int** main(){

**int** \*a, n;

    a = new **int**(n);

    std::cout << "Nhap do dai mang = ";

    std::cin >> n;

    Nhaparr(a, n);

    Xuatarr(a, n);

*// mảng giảm dần*

    std::cout  << "---mang giam dan--- ";

    mang\_giam\_dan\_di(a, n);

    Xuatarr(a, n);

}

m. Chèn số x vào mảng giảm dần sao cho mảng vẫn giảm dần sau khi chèn x.

#include <iostream>

**int** Nhap\_mang\_vao(**int** b[], **int** z){

    for (**int** i = 0; i < z; i++)

    {

        std::cout << "a[" << i << "] => ";

        std::cin >> b[i];

    }

}

**int** Xuat\_mang\_ra(**int** b[], **int** z){

    std::cout << "Mang = ";

    for (**int** i = 0; i < z; i++)

    {

        std::cout <<  b[i] << "; ";

    }

}

*// m. Chèn số x vào mảng giảm dần sao cho mảng vẫn giảm dần sau khi chèn x*

**int** chen\_x\_giam\_dan(**int** b[], **int** **&**z, **int** x, **int** vitri)

    {

**int** bien\_tam;

        for (**int** i = 0; i < z; i++)

        {

            for (**int** j = i + 1; j < z; j++)

            {

                if (b[i] < b[j])

                {

                    bien\_tam = b[i];

                    b[i] = b[j];

                    b[j] = bien\_tam;

                    for (**int** i = z - 1; i >= vitri; i--) *// n - 1 là thằng cuối cùng dịch đi trước*

*// i-- nó sẽ dịch xét từ n-1 tới về sau*

                    {

                        b[i + 1] = b[i]; *// a[i] = 4 thì a[i + 1] sẽ là 5 chúng ta sẽ gán từ phải qua trái tức là a[i] = a[i + 1] ;*

*// sau khi thêm nó sẽ nằm ở a[i];*

                    }

                    b[vitri] = x; *// thêm vị trí x vào bất kì trong mảng*

                    z++; *// sau k*

                }

            }

        }

    }

**int** main()

    {

**int** \*b, z;

        b = new **int**(z);

        std::cout << "Nhap do dai mang = ";

        std::cin >> z;

        Nhap\_mang\_vao(b, z);

        Xuat\_mang\_ra(b, z);

**int** x;

            std::cout << "Nhap phan tu can them = ";

            std::cin >> x;

**int** vitri;

            do

            {

                std::cout << "Nhap vi tri can them = ";

                std::cin >> vitri;

                if (vitri < 0 || vitri > z){

                    std::cout << "\n vi tri can them trong doan tu [0, " << z << "]";

                }

            } while (vitri < 0 || vitri > z);

            chen\_x\_giam\_dan(b , z, x, vitri);

            Xuat\_mang\_ra(b, z);

    }

n. Kiểm tra mảng có đối xứng hay không?

#include <iostream>

**int** Nhap\_mang\_A(**int** c[], **int** n)

{

    for (**int** i = 0; i <= n; i++)

    {

        std::cout << "a[" << i << "]";

        std::cin >> c[i];

    }

}

**int** Xuat\_mang\_ra\_ngoai(**int** c[], **int** n)

{

    std::cout << "Ta co mang = ";

    for (**int** i = 0; i <= n; i++)

    {

        std::cout << c[i] << "; ";

    }

}

**bool** kiem\_tra\_doi\_xung(**int** c[], **int** n){

    for (**int** i = 0; i <= n; i++)

    {

        if (c[i] == c[n  - i - 1])

        {

            return true;

            break;

        }

    }

    return false;

}

**int** xu\_ly\_doi\_xung(**int** c[], **int** n){

        if (kiem\_tra\_doi\_xung(c, n) == true)

        {

            std::cout << "Mang doi xung";

        }

        else

        {

            std::cout << "Mang kh doi xung";

        }

}

**int** main()

{

**int** \*c, n;

    c = new **int**(n);

    std::cout << "Nhap do dai mang = ";

    std::cin >> n;

    Nhap\_mang\_A(c, n);

    Xuat\_mang\_ra\_ngoai(c, n);

    xu\_ly\_doi\_xung(c, n);

}

**Bài 2**. Trộn 2 mảng tăng dần thành 1 mảng cũng tăng dần.

*// Trộn 2 mảng tăng dần thành 1 mảng cũng tăng dần.*

#include <iostream>

**int** Nhap\_arr\_A(**int** a[], **int** n){

    for (**int** i = 0; i <= n; i++)

    {

        std::cout << "A[" << i << "] => ";

        std::cin >> a[i];

    }

}

**int** Nhap\_arr\_B(**int** b[], **int** m){

    for (**int** i = 0; i <= m; i++)

    {

        std::cout << "B[" << i << "] => ";

        std::cin >> b[i];

    }

}

**int** Xuat\_arr\_A(**int** a[], **int** n)

{

    std::cout << "Mang A = ";

    for (**int** i = 0; i <= n; i++)

    {

        std::cout << a[i] << "; ";

    }

}

**int** Xuat\_arr\_B(**int** b[], **int** m){

    std::cout << "Mang B = ";

    for (**int** i = 0; i <= m; i++)

    {

        std::cout << b[i] << "; ";

    }

}

**void** doivitri(**int** **&**x, **int** **&**y){

    {

**int** bientam = x;

        x = y;

        y = bientam;

    }

}

**void** trontangdan(**int** arr[], **int** m, **int** n)

{

**int** mn = m + n + 1; *// mn la so luong phan tu trong mang arr tinh tu 0*

    for (**int** i = 0; i <= mn; i++) *// tao vong lap i tu 0 den mn*

    {

        for (**int** j = i + 1; j <= mn; j++) *// tao vong lap j tu i + 1 den mn*

        {

            if (arr[i] > arr[j]) *// neu a[i] > a[j] thi hoandoi*

            {

                doivitri(arr[i], arr[j]);

            }

        }

        std::cout << arr[i] << " "; *// in ra mang arr da sap xep tang dan*

    }

}

**int** main(){

**int** \*a, n;

    a = new **int**(n);

    std::cout << "Nhap do dai mang A = ";

    std::cin >> n;

**int** \*b, m;

    b = new **int**(m);

    std::cout << "Nhap do dai mang b = ";

    std::cin >> m;

    Nhap\_arr\_A(a, n);

    Xuat\_arr\_A(a, n);

    std::cout << "\n";

    Nhap\_arr\_B(b, m);

    Xuat\_arr\_B(b , m);

*// tron 2 mang vao nhau*

**int** arr[m + n + 2]; *// khai bao do dai mang tron m + n + 2 (vi mang a b bat dau tu 0 nen + 2 moi du)*

**int** x = 0; *// bien tam de luu so luong phan tu trong mang moi*

    for (**int** i = 0; i <= n; i++) *// vong lap de cho mang a vao mang arr*

    {

        arr[x] = a[i];

        x++;

    }

    for (**int** i = 0; i <= m; i++) *// vong lap de cho mang b vao mang arr*

    {

        arr[x] = b[i];

        x++;

    }

    std::cout << "=> mang sau khi tron: " << "\n";

    for (**int** i = 0; i < x; i++) *// vong lap de in ra mang arr*

    {

        std::cout << arr[i] << " ";

    }

    std::cout << "\n";

    std::cout << "=> mang tron da duoc sap xep tang dan: ";

    trontangdan(arr, m, n);

}

**Bài 3**. Nhập/xuất dữ liệu cho mảng int A[m][n] với m, n <= 10.

a. Nhập/xuất mảng A.

#include<iostream>

*//      a. Nhập/xuất mảng A.*

**int** a[100][100];

**int** m, n;

**void** NhapMaTran(**int** x[100][100], **int** **&**m, **int** **&**n){

    do

    {

        std::cout << "Nhap m = ";

        std::cin >> m;

        std::cout << "Nhap n = ";

        std::cin >> n;

    } while (n >= 10 || m >= 10);

    for (**int** i = 0; i < m; i++)

    {

        for (**int**  j = 0; j < n; j++)

        {

            std::cout << "a[" << i <<"]" << "[" << j << "] => ";

            std::cin >> a[i][j];

        }

    }

}

**void** XuatMaTran(**int** x[100][100], **int** **&**m, **int** **&**n)

{

    std::cout << "Ta cp mang hai chieu: ";

    for (**int** i = 0; i < m; i++)

    {

        for (**int** j = 0; j < n; j++)

        {

            std::cout << a[i][j] << ", ";

        }

    }

}

**int** main(){

    NhapMaTran(a, m, n);

    XuatMaTran(a, m ,n)

}

b. Có bao nhiêu phần từ số âm trong mảng.

#include<iostream>

*//      a. Nhập/xuất mảng A.*

**int** a[100][100];

**int** m, n;

**void** NhapMaTran(**int** x[100][100], **int** **&**m, **int** **&**n){

    do

    {

        std::cout << "Nhap m = ";

        std::cin >> m;

        std::cout << "Nhap n = ";

        std::cin >> n;

    } while (n >= 10 || m >= 10);

    for (**int** i = 0; i < m; i++)

    {

        for (**int**  j = 0; j < n; j++)

        {

            std::cout << "a[" << i <<"]" << "[" << j << "] => ";

            std::cin >> a[i][j];

        }

    }

}

**void** XuatMaTran(**int** x[100][100], **int** **&**m, **int** **&**n)

{

    std::cout << "Ta cp mang hai chieu: ";

    for (**int** i = 0; i < m; i++)

    {

        for (**int** j = 0; j < n; j++)

        {

            std::cout << a[i][j] << ", ";

        }

    }

}

*//     b. Có bao nhiêu phần từ số âm trong mảng.*

**int** kiem\_tra\_so\_am(**int** x[100][100], **int** **&**m, **int** **&**n)

{

**int** dem = 0;

    for (**int** i = 0; i < m; i++)

    {

        for (**int** j = 0; j < n; j++)

        {

            if (a[i][j] < 0){

                dem++;

            }

        }

    }

    std::cout << "so phan tu am trong mang la = " << dem << "\n";

}

**int** main(){

    NhapMaTran(a, m, n);

    XuatMaTran(a, m ,n);

    kiem\_tra\_so\_am(a, m, n);

}

c. Tìm giá trị lớn nhất trong A.

#include<iostream>

*//      a. Nhập/xuất mảng A.*

**int** a[100][100];

**int** m, n;

**void** NhapMaTran(**int** x[100][100], **int** **&**m, **int** **&**n){

    do

    {

        std::cout << "Nhap m = ";

        std::cin >> m;

        std::cout << "Nhap n = ";

        std::cin >> n;

    } while (n >= 10 || m >= 10);

    for (**int** i = 0; i < m; i++)

    {

        for (**int**  j = 0; j < n; j++)

        {

            std::cout << "a[" << i <<"]" << "[" << j << "] => ";

            std::cin >> a[i][j];

        }

    }

}

**void** XuatMaTran(**int** x[100][100], **int** **&**m, **int** **&**n)

{

    std::cout << "Ta cp mang hai chieu: ";

    for (**int** i = 0; i < m; i++)

    {

        for (**int** j = 0; j < n; j++)

        {

            std::cout << a[i][j] << ", ";

        }

    }

}

**int** gia\_tri\_lon\_nhat(**int** x[100][100], **int** **&**m, **int** **&**n){

**int** max = x[0][0];

    for (**int** i = 0; i < m; i++)

    {

        for (**int**  j = 0; j < n; j++)

        {

            if (max < a[i][j])

            {

                max = a[i][j];

            }

        }

    }

    std::cout << "Phan tu lon nhat trong mang la = "  << max << "\n";

}

**int** main(){

    NhapMaTran(a, m, n);

    XuatMaTran(a, m ,n);

   gia\_tri\_lon\_nhat(a, m ,n);

}

e. Sắp xếp từng dòng của A theo thứ tự tăng dần.

*//      e. Sắp xếp từng dòng của A theo thứ tự tăng dần*

#include<iostream>

*//      a. Nhập/xuất mảng A.*

**int** a[100][100];

**int** m, n;

**void** NhapMaTran(**int** x[100][100], **int** **&**m, **int** **&**n){

    do

    {

        std::cout << "Nhap m = ";

        std::cin >> m;

        std::cout << "Nhap n = ";

        std::cin >> n;

    } while (n >= 10 || m >= 10);

    for (**int** i = 0; i < m; i++)

    {

        for (**int**  j = 0; j < n; j++)

        {

            std::cout << "a[" << i <<"]" << "[" << j << "] => ";

            std::cin >> a[i][j];

        }

    }

}

**void** XuatMaTran(**int** x[100][100], **int** **&**m, **int** **&**n)

{

    std::cout << "Ta cp mang hai chieu: ";

    for (**int** i = 0; i < m; i++)

    {

        for (**int** j = 0; j < n; j++)

        {

            std::cout << a[i][j] << ", ";

        }

    }

}

**int** sap\_xep\_tang(**int** x[100][100], **int** **&**m , **int** **&**n){

**int** k = m \* n; *// tổng số lượng phần tử trong mảng*

    for (**int** i = 0; i < k - 1; i++)

    {

        for (**int** j = i + 1; j < k; j++)

        {

*// muốn biết đang ở dòng nào lấy i / n; n la số lượng cột*

            if (x[i / n][i % n] > x[j/n][j % n])

            {

**int** tam = x[i / n][i % n];

                x[i / n][i % n] = x[j / n][j % n];

                x[j / n][j % n] = tam;

            }

        }

    }

}

**int** main(){

    NhapMaTran(a, m, n);

    XuatMaTran(a, m ,n);

std::cout << "---Sap xep mang theo tang dan ---\n";

sap\_xep\_tang(a, m , n);

    XuatMaTran(a, m, n);

}

f. Sắp xếp mảng A theo thứ tự giảm dần.

#include<iostream>

*//      a. Nhập/xuất mảng A.*

**int** a[100][100];

**int** m, n;

**void** NhapMaTran(**int** x[100][100], **int** **&**m, **int** **&**n){

    do

    {

        std::cout << "Nhap m = ";

        std::cin >> m;

        std::cout << "Nhap n = ";

        std::cin >> n;

    } while (n >= 10 || m >= 10);

    for (**int** i = 0; i < m; i++)

    {

        for (**int**  j = 0; j < n; j++)

        {

            std::cout << "a[" << i <<"]" << "[" << j << "] => ";

            std::cin >> a[i][j];

        }

    }

}

**void** XuatMaTran(**int** x[100][100], **int** **&**m, **int** **&**n)

{

    std::cout << "Ta cp mang hai chieu: ";

    for (**int** i = 0; i < m; i++)

    {

        for (**int** j = 0; j < n; j++)

        {

            std::cout << a[i][j] << ", ";

        }

    }

}

**int** sap\_xep\_giam(**int** x[100][100], **int** **&**m, **int** **&**n)

{

**int** k = m \* n; *// tổng số lượng phần tử trong mảng*

    for (**int** i = 0; i < k - 1; i++)

    {

        for (**int** j = i + 1; j < k; j++)

        {

*// muốn biết đang ở dòng nào lấy i / n; n la số lượng cột*

            if (x[i / n][i % n] < x[j / n][j % n])

            {

**int** tam = x[i / n][i % n];

                x[i / n][i % n] = x[j / n][j % n];

                x[j / n][j % n] = tam;

            }

        }

    }

}

**int** main(){

    NhapMaTran(a, m, n);

    XuatMaTran(a, m ,n);

 std::cout << "---Sap xep mang theo giam dan---\n";

    sap\_xep\_giam(a, m, n);

    XuatMaTran(a, m ,n);

}

**Bài 4**. Cho ma trận A vuông cấp n (n<=10) với các phần tử là số nguyên:

a. Nhập/xuất A.

b. Tính tổng các phần tử nằm ngoài đường chéo chính.

#include <iostream>

**int** a[100][100];

**int**  n;

**void** NhapMaTran(**int** x[100][100],**int** **&**n)

{

    do

    {

        std::cout << "Nhap n = ";

        std::cin >> n;

    } while (n >= 10);

    for (**int** i = 0; i < n; i++)

    {

        for (**int** j = 0; j < n; j++)

        {

            std::cout << "a[" << i << "]" << "[" << j << "] => ";

            std::cin >> a[i][j];

        }

    }

}

**void** XuatMaTran(**int** x[100][100], **int** **&**n)

{

    std::cout << "Ta cp mang hai chieu: ";

    for (**int** i = 0; i < n; i++)

    {

        for (**int** j = 0; j < n; j++)

        {

            std::cout << a[i][j] << ", ";

        }

    }

}

**int** tong\_cheo\_chinh(**int** x[100][100], **int** **&**n){

**int** sum = 0; // mục tiêu là làm biến tạm để vô cộng các giá trị của mảng hai chiều

    for (**int** i = 0; i < n; i++)

    {

        for (**int** j = 0; j < n; j++)

        {

           if (i == j){

            sum+= a[i][j]; // sum + từng giá trị của mảng hai chiều

           }

        }

    }

    return sum;

}

**int** main(){

    NhapMaTran(a, n);

    XuatMaTran(a, n);

    std::cout << "\n";

    std::cout << "Tong duong cheo chinh = " << tong\_cheo\_chinh(a, n); // gọi hàm để xử lý

}

c. Tìm giá trị lớn nhất trên đường chéo chính.

#include <iostream>

**int** a[100][100];

**int** n;

**void** NhapMaTran(**int** x[100][100], **int** **&**n)

{

    do

    {

        std::cout << "Nhap n = ";

        std::cin >> n;

    } while (n >= 10);

    for (**int** i = 0; i < n; i++)

    {

        for (**int** j = 0; j < n; j++)

        {

            std::cout << "a[" << i << "]" << "[" << j << "] => ";

            std::cin >> a[i][j];

        }

    }

}

**void** XuatMaTran(**int** x[100][100], **int** **&**n)

{

    std::cout << "--Ta cp mang hai chieu-- \n";

    for (**int** i = 0; i < n; i++)

    {

        for (**int** j = 0; j < n; j++)

        {

            std::cout << a[i][j] << ", ";

        }

        std::cout << "\n";

    }

}

**int** gia\_tri\_lon\_nhat(**int** x[100][100], **int** **&**n)

{

**int** max = 0;

    for (**int** i = 0; i < n; i++)

    {

        for (**int** j = 0; j < n; j++)

        {

                if (max < a[i][j]){

                    max = a[i][j];

                }

        }

    }

    std::cout << max;

}

**int** main()

{

    NhapMaTran(a, n);

    XuatMaTran(a, n);

    std::cout << "\n";

    gia\_tri\_lon\_nhat(a, n);

}

d. Có bao nhiêu phần tử là số âm nằm trên đường chéo phụ.

#include <iostream>

**int** a[100][100];

**int** n;

**void** NhapMaTran(**int** x[100][100], **int** **&**n)

{

    do

    {

        std::cout << "Nhap n = ";

        std::cin >> n;

    } while (n >= 10);

    for (**int** i = 0; i <= n; i++)

    {

        for (**int** j = 0; j <= n; j++)

        {

            std::cout << "a[" << i << "]" << "[" << j << "] => ";

            std::cin >> a[i][j];

        }

    }

}

**void** XuatMaTran(**int** x[100][100], **int** **&**n)

{

    std::cout << "--Ta cp mang hai chieu-- \n";

    for (**int** i = 0; i <= n; i++)

    {

        for (**int** j = 0; j <= n; j++)

        {

            std::cout << a[i][j] << ", ";

        }

        std::cout << "\n";

    }

}

**int** so\_am\_trong\_duong\_cheo(**int** x[100][100], **int** **&**n)

{

**int** dem = 0; *// chéo phụ là bên phải*

    for (**int** i = 0; i <= n; i++)

    {

        for (**int** j = 0; j <= n; j++)

        {

            if (i + j == n){

                if (a[i][j] < 0){

                    dem++;

                }

            }

        }

    }

    std::cout << "So am trong mang = " << dem;

}

**int** main()

{

    NhapMaTran(a, n);

    XuatMaTran(a, n);

    std::cout << "\n";

    so\_am\_trong\_duong\_cheo(a, n);

}

e. Đếm số lượng số nguyên tố.

#include <iostream>

#include <cmath>

**int** a[100][100];

**int** n;

**void** NhapMaTran(**int** x[100][100], **int** **&**n)

{

    do

    {

        std::cout << "Nhap n = ";

        std::cin >> n;

    } while (n >= 10);

    for (**int** i = 0; i <= n; i++)

    {

        for (**int** j = 0; j <= n; j++)

        {

            std::cout << "a[" << i << "]" << "[" << j << "] => ";

            std::cin >> a[i][j];

        }

    }

}

**void** XuatMaTran(**int** x[100][100], **int** **&**n)

{

    std::cout << "--Ta cp mang hai chieu-- \n";

    for (**int** i = 0; i <= n; i++)

    {

        for (**int** j = 0; j <= n; j++)

        {

            std::cout << a[i][j] << ", ";

        }

        std::cout << "\n";

    }

}

**bool** kiem\_tra\_so\_nguyen\_to(**int** **&**n){

    if (n < 2){

        return false;

    }

    else {

        if (n ==  2){

            return false;

        }

        for (**int** i = 2; i <= sqrt(n); i++)

        {

            if (n % i == 0){

                return false;

            }

        }

        return true;

    }

}

**int** xet\_so\_nguyen\_to(**int** x[100][100], **int** **&**n)

{

**int** dem = 0;

    for (**int** i = 0; i <= n; i++)

    {

        for (**int** j = 0; j <= n; j++)

        {

           if (kiem\_tra\_so\_nguyen\_to(a[i][j]) == true){

                dem++;

           }

        }

    }

    std::cout << "so luong snt trong mang = " << dem;

}

**int** main()

{

    NhapMaTran(a, n);

    XuatMaTran(a, n);

    std::cout << "\n";

    xet\_so\_nguyen\_to(a, n);

}