**a. Khai báo một mảng số nguyên tối đa 30 phần tử.**

**b. Khởi tạo một mảng số nguyên với giá trị ngẫu nhiên (các giá trị nhỏ hơn 10).**

**Khai báo biến con trỏ p trỏ đến mảng vừa khởi tạo.**

**d. Dùng con trỏ và ký pháp độ dời để xuất các giá trị đang lưu trữ trong mảng.**

**e. Dùng con trỏ và ký pháp chỉ số để xuất các giá trị đang lưu trữ trong mảng.**

#include <iostream>

#include <iomanip>

using namespace std;

const int MAX = 30;

int main()

{

int a[MAX];

for (int i = 0; i < MAX; i++)

{

a[i] = rand() % 10;

}

int \*p;

p = a;

cout << "Con tro va ki phap do doi: " << endl;

for (int offset = 0; offset < MAX; offset++)

{

cout << \*(p + offset) << " ";

}

cout << "\ncon tro va ki phap chi so: " << endl;

for (int i = 0; i < MAX; i++)

cout << p[i] << " ";

cout << endl;

return 0;

}

**Viết chương trình dùng cấp phát động để cho phép 2 con trỏ lưu trữ giá trị của 2 số nguyên. Sau đó dựa trên địa chỉ mà con trỏ trỏ đến để tìm ước chung lớn nhất của 2 giá trị số nguyên này bằng thuật giải Euclid.**

#include <iostream>

#include <iomanip>

using namespace std;

int UCLN(int a, int b);

int main()

{

int \*p, \*q;

p = new int;

q = new int;

do

{

cout << "Nhap so nguyen thu nhat: ";

cin >> \*p;

} while (\*p <= 0);

do

{

cout << "Nhap so nguyen thu hai: ";

cin >> \*q;

} while (\*q <= 0);

cout << "UCLN cua " << \*p << " va " << \*q << " la: " << UCLN(\*p, \*q) << endl;

delete p, q;

return 0;

}

int UCLN(int a, int b)

{

if (b == 0)

return a;

return UCLN(b, a%b);

}

**Nhập vào một mảng số nguyên một chiều tối đa 30 phần tử. - Dùng con trỏ và ký pháp độ dời để xuất ra các phần tử là số lẻ trong mảng. - Trả về địa chỉ của giá trị phần tử lớn nhất trong mảng (giả sử các giá trị trong mảng không trùng nhau).**

#include <iostream>

#include <iomanip>

using namespace std;

const int MAX = 30;

void Input(int \*a, int n);

void Output(int \*a, int n);

int \*Maxa(int \*a, int n);

int main()

{

int n;

cout << "Nhap vao phan tu n: ";

cin >> n;

int \*a;

a = new int[n];

Input(a, n);

cout << "Cac phan tu le trong mang: " << endl;

Output(a, n);

int \*pMAX = Maxa(a, n);

cout << "Dia chi cua phan tu lon nhat " << \*pMAX << " la " << pMAX << endl;

return 0;

}

void Input(int \*a, int n)

{

cout << "Nhap vao mang so nguyen: ";

for (int i = 0; i < n; i++)

cin >> a[i];

}

void Output(int \*a, int n) // in cac phan tu le

{

for (int offset = 0; offset < n; offset++)

{

if (\*(a + offset) % 2 != 0)

cout << \*(a + offset) << " ";

}

cout << endl;

}

int \*Maxa(int \*a, int n) // Dia chi cua phan tu lon nhat

{

int max = a[0];

int \*p = NULL;

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

{

if (max < a[i])

{

max = a[i];

p = &a[i];

}

}

return p;

}

**Dùng con trỏ viết hàm xuất n số Fibonacci đầu tiên (n do người dùng nhập, là số nguyên > 1).**

#include <iostream>

#include <iomanip>

using namespace std;

void Fibo(int n);

int main()

{

int n;

cout << "Nhap vao so nguyen n: ";

cin >> n;

Fibo(n);

return 0;

}

void Fibo(int n)

{

int \*u, \*v, \*t;

u = new int;

v = new int;

t = new int;

\*u = 0;

\*v = 1;

if (n == 0)

return;

if (n == 1)

{

cout << "0\n";

return;

}

cout << "0 1";

int i = 2; // bien dem

while (i < n)

{

\*t = \*u + \*v;

cout << " " << \*t; // in so Fibo

\*u = \*v;

\*v = \*t;

i++;

}

delete u, v, t;

cout << endl;

}

**Dùng con trỏ viết hàm đảo ngược một mảng số nguyên một chiều**

#include <iostream>

#include <iomanip>

using namespace std;

const int MAX = 100;

void Input(int \*a, int n);

void Output(int \*a, int n);

void Daomang(int \*a, int n);

int main()

{

srand(time(0));

int n=0;

cout << "Nhap so nguyen n: ";

cin >> n;

int \*a;

a = new int[n];

Input(a, n);

cout << "Mang khoi tao ban dau: " << endl;

Output(a, n);

Daomang(a, n);

cout << "Mang sau khi dao nguoc la: " << endl;

Output(a, n);

delete[] a;

a = NULL;

return 0;

}

void Input(int \*a, int n)

{

for (int i = 0; i < n; i++)

a[i] = rand() % 20;

}

void Output(int \*a, int n)

{

for (int i = 0; i < n; i++)

{

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

}

cout << endl;

}

void Daomang(int \*a, int n)

{

for (int i = 0; i < n/2; i++)

{

int tmp = a[i];

a[i] = a[n - 1 - i];

a[n - 1 - i] = tmp;

}

}

**Dùng con trỏ viết hàm tìm kiếm một số nguyên trong mảng số nguyên một chiều. Nếu có thì trả về vị trí đầu tiên xuất hiện số nguyên đó. Nếu không thì trả về -1**

#include <iostream>

#include <iomanip>

using namespace std;

const int MAX = 100;

void Input(int \*a, int n);

void Output(int \*a, int n);

int Timkiem(int \*a, int n, int x);

int main()

{

srand(time(0));

int n = 0, x;

cout << "Nhap so nguyen n: ";

cin >> n;

int \*a;

a = new int[n];

Input(a, n);

cout << "Mang khoi tao ban dau: " << endl;

Output(a, n);

cout << "Nhap vao so can tim: ";

cin >> x;

int Vitri = Timkiem(a, n, x);

if (Vitri != -1)

cout << "Vi tri xuat hien la dai cua " << x << " la: " << Vitri +1 << endl;

else cout << "Khong tim thay!!!" << endl;

delete[] a;

a = NULL;

return 0;

}

void Input(int \*a, int n)

{

for (int i = 0; i < n; i++)

a[i] = rand() % 20;

}

void Output(int \*a, int n)

{

for (int i = 0; i < n; i++)

{

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

}

cout << endl;

}

int Timkiem(int \*a, int n, int x)

{

for (int i = 0; i < n; i++)

if (a[i] == x)

return i;

return -1;

}

**- Khai báo mảng hai chiều là một ma trận vuông cấp h (h >0).**

**- Lưu trữ giá trị của một tam giác Pascal với chiều cao h.**

**- In tam giác Pascal ra màn hình**

#include <iostream>

#include <iomanip>

using namespace std;

void Input(int \*\*a, int h); // Luu vao tam giac pascal

void Output(int \*\*a, int h); // Xuat tam giac Pascal

// ma tran vuong nen r=c goi chung la h

int main()

{

int h;

cout << "Nhap vao chieu cao h cua ma tran: ";

cin >> h;

int \*\*a;

a = new int\* [h];

// khoi tao cac phan tu = 0

for (int i = 0; i < h; i++)

a[i] = new int [h] {0};

Input(a, h);

cout << "Tam giac Pascal chieu cao " << h << " la: " << endl;

Output(a, h);

// don dep bo nho

for (int i = 0; i < h; i++)

delete[] a[i];

a = NULL;

return 0;

}

void Input(int \*\*a, int h)

{

for (int i = 0; i < h; i++)

{

for (int j = 0; j <= i; j++)

if (j == 0)

a[i][j] = 1;

else

a[i][j] = a[i - 1][j - 1] + a[i - 1][j];

}

}

void Output(int \*\*a, int h)

{

for (int i = 0; i < h; i++)

{

for (int j = 0; j <= i; j++)

cout << a[i][j] << " ";

cout << endl;

}

}

**- Hàm nhập một mảng số nguyên r hàng và c cột.**

**- Hàm xuất một mảng số nguyên r hàng c cột.**

**- Hàm dùng con trỏ để tính và trả về tổng các số nguyên tố trong mảng hai chiều.**

**- Hàm trả về các số nguyên tố trong mảng hai chiều.**

**- Hàm main để kiểm chứng.**

#include <iostream>

#include <iomanip>

using namespace std;

void Input(int \*\*a, int r, int c);

void Output(int \*\*a, int r, int c);

int TongSNT(int \*\*a, int r, int c);

bool KTSNT(int \*n);

void findPrimesInArray(int \*\*a, int r, int c);

int main()

{

int r, c;

cout << "Nhap vao hang va cot: ";

cin >> r >> c;

int \*\*a;

a = new int\* [r];

for (int i = 0; i < r; i++)

a[i] = new int [c];

Input(a, r, c);

cout << "Mang da luu tru: " << endl;

Output(a, r, c);

cout << "Cac so nguyen to trong mang: " << endl;

findPrimesInArray(a, r, c);

cout << "tong cac so nguyen to trong mang: " << TongSNT(a, r, c) << endl;

for (int i = 0; i < r; i++)

delete[] a[i];

delete[]a;

a = NULL;

return 0;

}

void Input(int \*\*a, int r, int c)

{

for (int i = 0; i < r; i++)

for (int j = 0; j < c; j++)

a[i][j] = rand() % 100;

}

void Output(int \*\*a, int r, int c)

{

for (int i = 0; i < r; i++)

{

for (int j = 0; j < c; j++)

cout << a[i][j] << " ";

cout << endl;

}

}

int TongSNT(int \*\*a, int r, int c)

{

int tong = 0;

int \*p = NULL;

for (int i = 0; i < r; i++)

for (int j = 0; j < c; j++)

{

if (KTSNT(&a[i][j]))

tong += a[i][j];

}

return tong;

}

bool KTSNT(int \*n)

{

if (\*n < 2) return false;

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

if (\*n%i == 0)

return false;

return true;

}

void findPrimesInArray(int \*\*a, int r, int c)

{

for (int i = 0; i < r; i++)

for (int j = 0; j < c; j++)

{

if (KTSNT(&a[i][j]))

cout << a[i][j] << " ";

}

cout << endl;

}

**Viết chương trình dùng cấp phát động để khởi tạo ngẫu nhiên giá trị của 2 mảng số nguyên theo dạng thức ma trận. Sau đó tính tổng và tích của 2 ma trận đó.**

#include <iostream>

#include <cstdlib> // thu vien chua ham rand()

#include <iomanip>

using namespace std;

// ham cap phat dong

int \*\*createMatrix(int r, int c)

{

int \*\*matrix = new int \*[r];

for (int i = 0; i < r; i++)

{

matrix[i] = new int[c];

}

return matrix;

}

// ham tao gia tri ngau nhien

void NgauNhien(int \*\*matrix, int r, int c)

{

for (int i = 0; i < r; i++)

{

for (int j = 0; j < c; j++)

matrix[i][j] = rand() % 10; // khoi tao gia tri ngau nhien tu 0 - 9

}

}

// ham tinh tong

int \*\*Tong\_Matrix(int \*\*matrix1, int \*\*matrix2, int r, int c)

{

int \*\*sum = createMatrix(r, c);

for (int i = 0; i < r; i++)

{

for (int j = 0; j < c; j++)

sum[i][j] = matrix1[i][j] + matrix2[i][j];

}

return sum;

}

// ham tinh tich

int \*\*Tich\_Matrix(int \*\*matrix1, int \*\*matrix2, int r1, int c1, int r2, int c2)

{

if (c1 != r2)

return NULL; // neu kich thuoc khong hop le tra ve NULL

int \*\*tich = createMatrix(r1, c2);

for (int i = 0; i < r1; i++)

{

for (int j = 0; j < c2; j++)

{

tich[i][j] = 0;

for (int k = 0; k < c1; k++)

tich[i][j] += matrix1[i][k] \* matrix2[k][j];

}

}

return tich;

}

// Hàm in ra ma trận

void printMatrix(int \*\*matrix, int r, int c)

{

for (int i = 0; i < r; i++)

{

for (int j = 0; j < c; j++)

cout <<setw(4)<< matrix[i][j] << " ";

cout << endl;

}

cout << endl;

}

int main()

{

int r1, c1, r2, c2;

cout << "Nhap kich thuoc cua ma tran 1: ";

cin >> r1 >> c1;

cout << "Nhap kich thuoc cua ma tran 2: ";

cin >> r2 >> c2;

// cap phat dong

int \*\*matrix1 = createMatrix(r1, c1);

int \*\*matrix2 = createMatrix(r2, c2);

NgauNhien(matrix1, r1, c1);

NgauNhien(matrix2, r2, c2);

cout << "Ma tran 1: " << endl;

printMatrix(matrix1, r1, c1);

cout << "Ma tran 2: " << endl;

printMatrix(matrix2, r2, c2);

// tong va tich 2 ma tran

int \*\*sum = Tong\_Matrix(matrix1, matrix2, r1, c1);

int \*\*tich = Tich\_Matrix(matrix1, matrix2, r1, c1, r2, c2);

// kiem tra kich thuoc co hop le hay khong

//tong 2 ma tran

if (r1 != r2 || c1 != c2)

cout << "khong the cong hai ma tran nay." << endl;

else

{

cout << "Tong cua hai ma tran la: " << endl;

printMatrix(sum, r1, c1);

}

// tich 2 ma tran

if (c1 != r2)

cout << "Khong the nhan hai ma tran nay." << endl;

else

{

cout << "Tich cua hai ma tran la: " << endl;

if (tich == NULL)

cout << "Khong the nhan hai ma tran nay." << endl;

else

printMatrix(tich, r1, c2);

}

// Giai phong bo nho

for (int i = 0; i < r1; i++)

delete[] matrix1[i];

delete[] matrix1;

for (int i = 0; i < r2; i++)

delete[] matrix2[i];

delete[] matrix2;

for (int i = 0; i < r1; i++)

delete[] sum[i];

delete[] sum;

for (int i = 0; i < r1; i++)

delete[] tich[i];

delete[] tich;

return 0;

}