﻿//Bài 0 Viết phương trình bậc 1 ax + b = 0

//Bài 1.1 Viết hàm giải phương trình bậc 2 có dạng : ax ^ 2 + bx + c = 0

//Bài 1.2 Viết hàm giải phương trình bặc 3 có dạng : ax^3 + bx^2 + cx + d = 0

//Bài 1.3 Viết chương trình nhận vào năm.Xuất ra màn hình thông báo năm đó là năm nhuần hay năm không nhuần ?

//Bài 1.4 Viết chương trình nhận vào tháng.Xuất ra màn hình số ngày của tháng đó.

//Bài 1.5 Viết chương trình có chức năng như máy tính cầm tay(sử dụng switch)

//Bài 1.5 (NÂNG CAO) Viết chương trình có giao diện là một menu lựa chọn 1 trong 4 phép tính(+-\*/ )

// và lựa chọn thoát. Khi người dùng chọn lựa chọn nào sẽ xử lý tương ứng.

// + Với 4 phép tính thì cho người dùng nhập tiếp 2 số nguyên a và b.

// Thực hiện phép tính đã chọn với 2 số đó (lưu ý lỗi chia 0 và kết quả phép chia phải làm tròn lấy 2 chữ số thập phân).

// + Với lựa chọn thoát thì sẽ kết thúc chương trình.

// + Sau khi lựa chọn và tính 1 kết quả phép tính tương ứng, chương trình thể hiện

// lại menu cho người dùng tiếp tục có lựa chọn phép tính hay thoát để kết thúc.

//Bài 1.6 Viết chương trình nhận vào 3 số nguyên. Xuất ra màn hình 3 số đó được sắp xếp tăng dần

//Bài 1.7 Viết chương trình nhận vào số đo 3 cạnh của 1 tam giác. Kiểm tra xem 3 cạnh đó có hợp lệ hay không?

// Nếu hợp lệ thì tam giác đó là loại tam giác gì (đều, vuông,cân, vuông cân hay thường) ?

//Bài 1.8 Viết chương trình nhận vào 1 số nguyên n(n > 0).

// Tính và xuất ra màn hình kết quả tổng các số từ 1 đến n.

//Bài 1.9 Viết chương trình nhận vào 1 số nguyên dương n > 1. Tính và xuất tích các số lẻ trong phạm vi từ 1 đến n.

//Bài 1.10 Tính và xuất ra màn hình kết quả tổng các số lẻ và tổng các số chẵn trong phạm vi từ 1 đến n

//Bài 1.11 Viết chương trình nhận vào 1 số nguyên dương n.Xuất ra màn hình số chữ số và

// tổng các chữ số của n.Ví dụ : n = 3523 thì kết quả là n có 4 chữ số và tổng các chữ số là 13.

//Bài 1.12 Viết chương trình nhận vào 1 số nguyên dương n.Xuất ra màn hình số đảo ngược của n.

// Ví dụ n là 352 thì số đảo ngược của nó là 253.

// Lưu ý : thực hiện việc kiểm tra giá trị n ban đầu nhập vào.Nếu n <= 0 thì thông

// báo nhập sai và cho phép nhập lại cho đến khi nào n > 0 như đề toán yêu cầu thì

// mới dừng việc nhập để tính toán.

//Bài 1.13 Viết chương trình nhận vào 1 số nguyên n. Xuất ra màn hình thông báo n có phải là số nguyên tố hay không ?

// Biết rằng số nguyên tố là số từ 2 trở lên, chỉ chia hết cho 1 và cho chính bản thân nó.

//Bài 1.13 (NÂNG CAO) Viết chương trình nhận vào 2 số nguyên dương n1 và n2(n1 < n2).

// Đếm và xuất kết quả xem từ n1 đến n2 có bao nhiêu số là số nguyên tố ?

// Biết rằng số nguyên tố là số từ 2 trở lên, chỉ chia hết cho 1 và chính bản thân nó.

//Bài 1.14 Viết chương trình nhận vào 1 số nguyên n > 0. Tính và xuất tổng các chữ số lẻ của n.

//Vd : n= 12345 thì tong=9

//Bài 1.15 Viết chương trình nhận vào 2 số nguyên dương n1 và n2(n1 < n2).

// Đếm và xuất ra màn hình kết quả có bao nhiêu số là ước của 10 trong phạm vi từ n1 đến n2.

//Bài 1.15 (NÂNG CAO) Viết chương trình nhận vào 2 số nguyên dương n1 và n2(n1 < n2).

//Đếm xem từ n1 đến n2 có bao nhiêu số chẵn ? bao nhiêu số lẻ ? bao nhiêu số là ước của 10 ?

//Bài 1.16 Viết chương trình nhận vào 1 số nguyên n > 0. Xuất ra màn hình các giá trị thừa

// số nguyên tố của n.Ví dụ n = 126 thì các thừa số nguyên tố là 2, 3, 3, 7.

//Bài 1.17 Viết chương trình nhận vào 1 số ở dạng nhị phân.Xuất ra màn hình kết quả số đó chuyển qua hệ thập phân.

//Bài 1.18 Viết chương trình nhận vào 1 số nguyên n > 0. Tách từng chữ số của n và thể hiện chúng dưới dạng cách nhau một khoảng cách.

// Ví dụ n = 3456 thì kết quả từng chữ số là 3 4 5 6

//Bài 1.19 Viết chương trình cho người dùng nhập giá trị cho m số nguyên. Tính và xuất

//tổng các số vừa nhập.Giao diện chương trình khi thực hiện được mô phỏng ở ví dụ sau :

//Ban can bao nhieu so : 3

//Nhap so thu 1 : 6

//Nhap so thu 2 : 4

//Nhap so thu 3 : 2

//Tong 3 so vua nhap la : 12

//Bài 1.20 Viết chương trình nhận vào 1 số nguyên dương n. Tính và xuất kết quả tính n giai thừa.

//Bài 1.21 Viết chương trình nhận vào 2 số nguyên dương là chiều dài d và chiều rộng r của hình chữ nhật.

// Xuất ra hình chữ nhật dưới dạng các dấu \* .

// Ví dụ d = 3, r = 5 thì kết quả :

// \*\*\*\*\*

// \*\*\*\*\*

// \*\*\*\*\*

//Bài 1.22 Viết chương trình nhận vào chiều cao h của một hình tam giác(h > 0).

// Xuất ra màn hình tam giác có chiều cao h dưới dạng các dấu\* .

// Ví dụ h = 3 thì kết quả :

// \*

// \*\*

// \*\*\*

//Bài 1.23 Viết chương trình nhận vào 1 số nguyên dương n(n > 0).

// Tính và xuất tổng các số lẻ từ 1 đến n.Ví dụ : n = 4 thì kết quả là 4.

//Bài 1.24 Viết chương trình nhận vào 1 số nguyên dương n(n > 0).

// Xuất ra màn hình bảng cửu chương ứng với số nguyên n.Ví dụ n = 3 thì kết quả xuất ra màn hình :

// 3 x 1 = 3

// 3 x 2 = 6

// …

// 3 x 10 = 30

//Bài 1.25 Viết chương trình nhận vào 1 số nguyên dương c là chiều dài cạnh của hình

// vuông(c > 0). Xuất ra hình vuông dưới dạng các dấu \* .

// a.Hình vuông với n = 4

// \*\*\*\*

// \*\*\*\*

// \*\*\*\*

// \*\*\*\*

// b.Hình vuông với n = 4

// \*\*\*\*

// \* \*

// \* \*

// \*\*\*\*

//Bài 1.26 Viết chương trình nhận vào chiều cao h của một hình tam giác(h > 0).Xuất ra

// màn hình tam giác có chiều cao h dưới dạng các dấu\* .

// a.Tam giác với h = 5:

// \*

// \*\*

// \* \*

// \* \*

// \*\*\*\*\*

// b.Tam giác với h = 4

// \*

// \*\*

// \*\*\*

// \*\*\*\*

// c.Tam giác với h = 4

// \*\*\*\*

// \*\*\*

// \*\*

// \*

// d.Tam giác với h = 4

// \*\*\*\*

// \*\*\*

// \*\*

// \*

// e.Tam giác với h = 4

// \*

// \* \*

// \* \* \*

// \* \* \* \*

//Bài 1.27 Viết các hàm sau đây và chương trình kiểm tra các hàm đã viết :

// a.Hàm tính diện tích hình chữ nhật.

// b.Hàm tính chu vi hình chữ nhật

//Bài 1.28 Viết các hàm sau đây và chương trình kiểm tra các hàm đã viết :

// a.Hàm tìm số lớn nhất giữa 2 số nguyên.

// b.Hàm tìm số lớn nhất giữa 3 số nguyên.

// Xây dựng hàm tìm số lớn nhất của hai số nguyên.

//Bài 1.29 Viết hàm kiểm tra một số nguyên n có phải là số hoàn thiện hay không ? Viết chương

// trình kiểm tra hàm vừa viết.Biết rằng số hoàn thiện là số có tổng các ước số(kể cả 1)

// bằng chính nó.Ví dụ : 6 là số hoàn thiện vì 6 = 1 + 2 + 3 (1, 2, 3 là các ước của 6).

//Bài 1.30 Viết hàm kiểm tra một số nguyên n có phải là số đối xứng(Palindrome) hay không ?

// Viết chương trình kiểm tra hàm vừa viết.Biết rằng số đối xứng là số có số đảo ngược

// của nó bằng chính nó.

//Bài 1.31 Viết các hàm sau đây và chương trình kiểm tra các hàm đã viết :

// a.Hàm tính n!(n là số nguyên không âm).

// b.Hàm tính tổ hợp n chập k theo công thức

// c.Hàm in tam giác Pascal có chiều cao h

//Bài 2. Nhập vào số nguyên dương n(n > 0).Viết hàm xuất tất cả các chữ số là số nguyên tổ của số nguyện dương n

//Bài 3. Nhập vào số nguyên dương n(n > 0).Việt hàm kiểm tra xem n có phải là số hoàn thiện không ?

//Bài 4. Nhập vào số nguyên dương n(n > 0).Viết hàm kiểm tra xem n có phải là số đối xứng hay không ?

//Bài 5 : Nhập vào số nguyên dương n(n > 0).Viết hàm trình tính : S(n) = i ^ 2 + 2 ^ 2 + 3 ^ 2 + . + n ^ 2

//Bài 6 : Viết hàm in ra tam giác vuông cân với chiều cao h

//Ví dụ : h = 4

//Bài 7 : Viết hàm tìm các số nguyên tố nhỏ hơn một số nguyên dương n cho trước(n > 0).

//vd : Nhập số nguyên dương : 11

//Các số nguyên tổ nhỏ hơn 11 là : 2357

//Bài 8.1 Viết chương trình thực hiện các công việc sau đây :

// a) Khai báo một mảng alpha lưu 50 phần tử kiểu double.

// b) Gán giá trị của 25 phần tử đầu tiên là bình phương của chỉ số phần tử, 25 phần

// tử còn lại có giá trị là lập phương của chỉ số.

// c) In các giá trị trong mảng alpha ra màn hình, mỗi dòng gồm 10 phần tử.

//Bài 8.2 Viết chương trình thực hiện các công việc sau đây :

// a) Định nghĩa hằng MAXSIZE là 100.

// b) Khai báo mảng values có thể lưu số phần tử tối đa là MAXSIZE, kiểu của mỗi

// phần tử là double.

// c) Khai báo currentSize lưu số phần tử hiện có trong mảng.

// d) Nhập vào giá trị của currentSize.

// e) Nhập vào các giá trị trong mảng values, số phần tử đọc vào là currentSize.

// f) In các phần tử trong mảng values trên một dòng, cách nhau bởi dấu phẩy.

//Bài 8.3 Viết chương trình nhập vào một mảng số nguyên(tối đa 100 phần tử), tìm và in ra

//phần tử lớn nhất của mảng.

//Bài 8.4 Viết chương trình nhập vào một mảng số nguyên(tối đa 100 phần tử), tính và in ra

//tổng và giá trị trung bình của các phần tử trong mảng.

//Bài 8.5 Viết chương trình nhập vào một mảng số nguyên(tối đa 100 phần tử) và vị trí của

//phần tử cần xóa, xóa phần tử tại vị trí đã nhập và in ra mảng sau khi xóa.

//Bài 8.6 Viết chương trình gồm các hàm sau đây:

// a) Nhập một dãy số nguyên vào mảng arr, số phần tử nhập vào là arrSize :

// void nhap(int arr[], int arrSize);

// b) In dãy số nguyên trong arr, cách nhau bởi dấu phẩy, arrSize là số phần tử của

// dãy :

// void xuat(const int arr[], int arrSize);

// c) Tìm một giá trị x trong mảng arr, trả về true tìm thấy, ngược lại trả về false,

// arrSize là số phần tử của mảng :

// bool tim(const int arr[], int arrSize, int x);

// d) Đảo ngược các phần tử trong mảng arr, arrSize là số phần tử của mảng :

// void daonguoc(int arr[], int arrSize)

// e) Xóa phần tử tại một vị trí p trong mảng arr, arrSize là số phần tử của mảng :

// void xoa(int arr[], int& arrSize, int p)

// f)Thêm phần tử vào cuối mảng

// k)Thêm phần tử vào đầu mảng

//Bài 8.11 Viết hàm sắp xếp các phần tử trong mảng tăng dần

//Bài 8.12 Viết hàm xóa các phần tử trùng nhau trong mảng

//Bài 1.32 Viết hàm kiểm tra 1 giá trị năm nhận vào có phải là năm nhuần hay không.Viết

//chương trình kiểm tra lại hàm vừa xây dựng.

//Bài 1.33 Viết hàm tìm ước chung lớn nhất của 2 số nguyên dương bằng thuật giải Euclid.

// Thuật giải Euclid :

// -Nhận vào : 2 số nguyên dương m và n.

// - Trả về : ước chung lớn nhất của m và n.

// - Thuật giải :

// o Bước 1 : nếu n = 0 thì kết quả là m và kết thúc.Ngược lại sang bước 2.

// o Bước 2 : tính số dư r của phép chia m cho n.

// o Bước 3 : gán n cho m, r cho n.Quay lại bước 1.

// Viết chương trình nhận vào 1 phân số(gồm tử số và mẫu số), xuất kết quả rút gọn

// của phân số(dùng hàm ước chung lớn nhất vừa xây dựng).

//Bài 1.34 Viết các hàm sau đây :

// a.Hàm kiểm tra một ký tự có phải là ký tự số hay không ?

// b.Hàm kiểm tra một ký tự có phải là ký tự chữ cái hay không ?

// c.Chương trình nhận vào một chuỗi(string), xuất kết quả chuỗi vừa nhập có bao

// nhiêu ký tự số ? bao nhiêu ký tự chữ ? Lưu ý : sử dụng 2 hàm đã xây dựng ở trên.

//Bài 1.35 Viết các hàm sau đây :

// a.Hàm kiểm tra một ký tự có phải là ký tự in hoa hay không ?

// b.Hàm kiểm tra một ký tự có phải là ký tự in thường hay không ?

// c.Chương trình nhận vào một chuỗi(string), xuất kết quả chuỗi vừa nhập có bao

// nhiêu ký tự in hoa ? bao nhiêu ký tự in thường ? Lưu ý : sử dụng 2 hàm đã xây

// dựng ở trên

//Bài 1.36 Viết hàm nhận vào 3 số ngày, tháng, năm; trả về số ngày cách ngày đầu tiên trong

//năm.Viết chương trình kiểm tra hàm vừa viết.

//Ví dụ :

//3 giá trị ngày - tháng - năm là 1 - 1 - 2006 thì kết quả là 1.

//3 giá trị ngày - tháng - năm là 25 - 12 - 2006 thì kết quả là 359

//Đề kt thử Câu 1) Khai báo nguyên mẫu hàm và định nghĩa các hàm sau:

// a) Hàm S(n) nhận vào một số nguyên dương n(n > 0) và trả về trị của đa thức :

// S(n) = 1 + 1 / 2 - 1 / 3 + 1 / 4 – 1 / 5 … + 1 / n(1, 5 điểm)

// b) Hàm SoChuSoLe(n) : nhận vào số nguyên dương n(n > 0) và trả về số các chữ số lẻ có

// trong n. (1, 5 điểm)

// c) Áp dụng hai hàm trên, viết chương trình hiển thị một menu cho phép người dùng chọn

// một trong hai tác vụ(1 hoặc 2), sau đó cho phép nhập số nguyên dương n và hiển thị kết

// quả ra màn hình.Chương trình hiển thị yêu cầu hỏi người dùng có tiếp tục hay không,

// nếu muốn tiếp tục, nhấn ký tự ‘y’ hoặc ‘Y’, màn hình được xóa và hiển thị lại menu cho

// phép người dùng tiếp tục thực hiện thao tác, nếu nhập ký tự bất kỳ khác, chương trình sẽ

// kết thúc(2 điểm)

//Đề kt thử Câu 2) Viết chương trình cho phép tạo ngẫu nhiên một mảng số nguyên n phần tử (n được nhập

// từ bàn phím), các số trong mảng chỉ chứa các giá trị từ 1 – 20. (1 điểm)

// a) In các giá trị trong mảng ra màn hình, mỗi phần tử cách nhau dấu phẩy và 1 khoảng trắng (1 điểm)

// b) Xuất kết quả thống kê cho biết mảng có bao nhiêu số chẵn và trung bình cộng các số

// chẵn trong mảng. (1, 5 điểm)

// c) Sắp xếp mảng theo thứ tự : các số lẻ nằm về bên trái có thứ tự giảm dần, các số chẵn nằm

// về bên phải có thứ tự tăng dần(lưu ý : không dùng mảng phụ).In mảng kết quả ra màn

// hình. (1, 5 điểm)

// Ví dụ

// Mảng ban đầu : 6, 5, 8, 12, 17, 8, 4, 3, 1, 5

// Mảng đã sắp xếp : 17, 5, 5, 3, 1, 4, 6, 8, 8, 12

//Bài 1.37 Viết hàm cho biết ngày hôm trước của một ngày cho trước

//Bài 1.38 Viết hàm cho biết ngày hôm sau của một ngày cho trước

//Bài 1.39 Tính ước chung lớn nhất giữa a và b

//Bài 1.40 Chuyển đổi giữa hệ 16 8 10

//Bài 1.41 Chuyển đổi hệ 10 sang hệ 2

//Bài 0 Viết phương trình bậc 1 ax + b = 0

// Phụ lục :

//\* a=0 b!=0 => Phương trình vô nghiệm

//\* a=0 b=0 => Phương trình vô số nghiệm

//\* a!=0 b!=0 => Phương trình có nghiệm duy nhất

#include<iostream>

#include<iomanip> // setw ; fixed << setprecision

using namespace std;

int main0()

{

double x, a, b;

cout << "GIAI PHUONG TRINH BAC NHAT \nax + b = 0\n";

cout << setfill('-'); //Chuyển chế độ mặt định setw " " thành setw "-"

cout << setw(55) << "-" << endl;

cout << "Nhap gia tri a va b cho phuong trinh bac nhat: ";

cin >> a >> b;

if (a == 0)

{

if (b == 0)

cout << "Phuong trinh vo so nghiem !";

else // b!= 0

cout << "Phuong trinh vo nghiem !";

}

else // a != 0

{

x = (double)-b / a;

cout << "Phuong trinh co nghiem duy nhat x bang : " << fixed << setprecision(2) << x;

} // fixed << setprecision(2) lấy 2 số dư

cout << endl;

system("pause");

return 0;

}

//Bài 1.1 Viết hàm giải phương trình bậc 2 có dạng : ax ^ 2 + bx + c = 0

// Phụ lục

// Ẩn: x1,x2,a,b,c,denta

// 1/ a==0 p.trinh trở thành p.trình bậc 1

// a/ a==0 b==0 c==0 phương trình vô số nghiệm

// a==0 b==0 c!=0 phương trình vô nghiệm

// b/ a==0 b!=0 phương trình bậc 1 có nghiệm duy nhất x=-c/b

// 2/ a!=0 vẫn là phương trình bậc 2

// denta=b\*b-4ac

// a/ denta == 0 phương trình có nghiệm kép x1=x2=-b/2a

// b/ denta <0 phương trình vô nghiệm

// c/ denta >0 phương trình có 2 nghiệm phân biệt x1=(-b + sqrt(denta))/2a x2=(-b - sqrt(denta))/2a

// sqrt() : là căn bậc 2 của 1 số

#include<iostream>

using namespace std;

#include<iomanip>; // dùng setw() và setfill, fixed<<setprecision(2)

int main1()

{

double x1, x2, a, b, c, denta;

cout << "GIAI PHUONG TRINH BAC 2 ax^2 + bx + c =0\n";

cout << setfill('-'); //chuyển chế độ mặc định set() " " thành setw() "-"

cout << setw(60) << "-" << endl;

cout << "Nhap he so lan luot cho a, b,c tai day: ";

cin >> a >> b >> c;

if (a == 0)

{

cout << "Vi a bang 0 nen phuong trinh ve bac 1";

if (b == 0)

{

if (c == 0)

cout << "Phuong trinh co vo so nghiem !";

else // c!=0

cout << "Phuong trinh vo nghiem !";

}

else // b!=0

cout << "Phuong trinh co nghiem duy nhat la : x = " << -c / b;

}

else //a!=0

{

denta = b \* b - 4 \* a \* c;

if (denta < 0)

cout << "Phuong trinh vo nghiem !";

if (denta == 0)

cout << fixed << setprecision(2) << "Phuong trinh co nghiem kep x1=x2=" << -b / 2 \* a;

else // denta > 0

{

x1 = (-b + sqrt(denta)) / (2 \* a);

x2 = (-b - sqrt(denta)) / (2 \* a);

cout << fixed << setprecision(2) << "Phuong trinh co hai nghiem phan biet x1,x2 lan luot la x1=" << x1 << " ,x2=" << x2;

}

}

system("pause");

return 0;

}

//Bài 1.2 Viết hàm giải phương trình bặc 3 có dạng : ax^3 + bx^2 + cx + d = 0

//Phụ lục

//Ẩn: x1,x2,x3,denta,a,b,c,d,k

// 1/ a==0 p.trình trở thành phương trình bậc 2

// a/ a==0 b==0 trở thành phương trình bậc 1

// \* a==0 b==0 c==0 d==0 phương trình vô số nghiệm

// a==0 b==0 c==0 d!=0 phương trình vô nghiệm

// \*\* a==0 b==0 c!=0 phương trình có nghiệm duy nhất

// b/ a==0 b!=0 vẫn là phương trình bậc 2

// denta=b^2 -4ac

// \* denta < 0 phương trình vô nghiệm

// \*\* denta == 0 phương trình có nghiệm kép x1=x2=-b/2a

// \*\*\* denta > phương trình có 2 nghiệm phân biệt x1=(-b + sqrt(denta))/2a x2=(-b - sqrt(denta))/2a

// sqrt() : là căn bậc 2 của 1 số

// 2/ a!=0 vẫn là phương trình bậc 3

// denta = b\*b -3\*a\*c

// k = (9\*a\*b\*c - 2\*pow(b,3)-27\*pow(a,2)\*d) / (2\*sqrt(pow(abs(denta),3 )))

// a/ denta > 0

// \* abs(k) <= 1 Phương trình có 3 nghiệm

// x1 = (2\*sqrt(denta)\*cos(acos(k)/3)-b)/(3\*a)

// x2 = (2\*sqrt(denta)\*cos(acos(k)/3-(2\*PI/3))-b)/(3\*a)

// x3 = (2\*sqrt(denta)\*cos(acos(k)/3+(2\*PI/3))-b)/(3\*a)

// nhớ # define PI 3.1415926535898

// \*\* abs(k) > 1 : Phương trình có một nghiệm duy nhất

// x = ((sqrt(denta)\*abs(k))/(3\*a\*k))\*(pow(abs(k)+sqrt(k\*k-1),1/3)+pow(abs(k)-sqrt(k\*k-1),1/3))-(b/3\*a)

// b/ denta = 0 : Phương trình có một nghiệm bội

// x = (-b + pow(b\*b\*b-27\*a\*a\*d , 1/3)) / 3\*a

// c/ denta < 0 : Phương trình có một nghiệm duy nhất

// x = (sqrt(abs(denta))/3\*a)\*((pow(k+sqrt(k\*k+1),1/3)+ pow(k - sqrt(k\*k+1) ,1/3 ))-(b/3\*a)

#include<iostream>

#include<iomanip> // dùng setw() và setfill, fixed<<setprecision(2)

#include<cmath> // có tính toán và dùng hàm sqrt() pow( , ) acos( )

using namespace std;

void Phuongtrinhbac2(double, double, double);

void Phuongtrinhbac1(double, double);

void Phuongtrinhbac3(double, double, double, double);

# define PI 3.1415926535898

int main2()

{

double a, b, c, d;

cout << "GIAI PHUONG TRINH BAC 3 \nax^3 + bx^2 + cx + d = 0\n";

cout << setfill('-');

cout << setw(60) << "-" << endl;

cout << "Nhap he so lan luot cho a,b,c,d : ";

cin >> a >> b >> c >> d;

if (a == 0)

{

cout << "a==0 phuong trinh tro ve bac 2 : bx^2 + cx +d = 0\n";

if (b == 0)

{

cout << "a==0 && b==0 phuong trinh tro ve bac 1 : cx + d = 0\n";

Phuongtrinhbac1(c, d);

}

else // b!=0

{

cout << "Giai phuong trinh bac 2 : bx^2 + cx +d = 0\n";

Phuongtrinhbac2(b, c, d);

}

}

else //a!=0

{

//Phuongtrinhbac3( a, b, c, d) ;

}

system("pause");

return 0;

}

void Phuongtrinhbac2(double b, double c, double d) // 1 2 3 // bx^2 + cx +d = 0

{

double denta, x1, x2;

denta = c \* c - 4 \* b \* d; // 2\*2 - 4\*1\*3

if (denta == 0)

cout << "Phuong trinh co nghiem kep x1=x2=" << -c / 2 \* b << endl;

if (denta < 0)

cout << "Phuong trinh vo nghiem\n";

else// denta > 0

{

x1 = (-c + sqrt(denta)) / (2 \* b);

x2 = (-c - sqrt(denta)) / (2 \* b);

cout << fixed << setprecision(2) << "Phuong trinh co 2 nghiem phan biet x1=" << x1 << " , x2=" << x2 << endl;

}

}

void Phuongtrinhbac1(double c, double d) // cx + d = 0

{

if (c == 0)

{

if (d == 0)

cout << "Phuong trinh co vo so nghiem !\n";

else // d!=0

cout << "Phuong trinh vo nghiem !\n";

}

else // c!=0

{

cout << fixed << setprecision(2) << "Phuong trinh co nghiem duy nhat x=" << -d / c << endl;

}

}

void Phuongtrinhbac3(double a, double b, double c, double d)

{

double denta, k, x1, x2, x3;

denta = b \* b - 3 \* a \* c;

k = (9 \* a \* b \* c - 2 \* pow(b, 3) - 27 \* pow(a, 2) \* d) / (2 \* sqrt(pow(abs(denta), 3)));

if (denta > 0)

{

if (k <= 1)

{

x1 = (2 \* sqrt(denta) \* cos(acos(k) / 3) - b) / (3 \* a);

x2 = (2 \* sqrt(denta) \* cos(acos(k) / 3 - (2 \* PI / 3)) - b) / (3 \* a);

x3 = (2 \* sqrt(denta) \* cos(acos(k) / 3 + (2 \* PI / 3)) - b) / (3 \* a);

cout << fixed << setprecision(2) << "Phuong trinh co 3 nghiem phan biet x1=" << x1 << " , x2=" << x2 << " , x3=" << x3;

}

else // k > 1

{

cout << fixed << setprecision(2) << "Phuong trinh co mot nghiem duy nhat x=" << ((sqrt(denta) \* abs(k)) / (3 \* a \* k)) \* (pow(abs(k) + sqrt(k \* k - 1), 1 / 3) + pow(abs(k) - sqrt(k \* k - 1), 1 / 3)) - (b / 3 \* a);

}

}

if (denta == 0)

{

cout << fixed << setprecision(2) << "Phuong trinh co mot nghiem boi x=" << (-b + pow(b \* b \* b - 27 \* a \* a \* d, 1 / 3)) / 3 \* a;

}

else // denta < 0

{

cout << fixed << setprecision(2) << "Phuong trinh co mot nghiem duy nhat x=" << (sqrt(abs(denta)) / (3 \* a)) \* ((pow(k + sqrt(k \* k + 1), 1 / 3) + pow(k - sqrt(k \* k + 1), 1 / 3))) - (b / (3 \* a));

}

}

//Bài 1.3 Viết chương trình nhận vào năm.Xuất ra màn hình thông báo năm đó là năm nhuần hay năm không nhuần ?

//Biết rằng năm nhuần là năm có 366 ngày.

//Năm nhuần còn là năm chia hết cho 400 hoặc chia hết cho 4 nhưng không chia hết cho 100

//Nhớ kiểm tra năm nhập vào có hợp lệ ko : năm > 0 là hợp lệ

#include<iostream>

using namespace std;

int main3()

{

int n;

cout << "KIEM TRA NAM NHUAN HAY KHONG NHUAN\n";

cout << "Nhap nam vao de kiem tra : ";

cin >> n;

if (n > 0)

{

if (n % 400 == 0 || (n % 4 == 0 && n % 100 != 0))

cout << "Nam nhap vao " << n << " la nam nhuan !\n";

else

cout << "Nam nhap vao " << n << " la nam khong nhuan !\n";

}

else // n <= 0

cout << "Nam nhap vao khong hop le !\n";

system("pause");

return 0;

}

//Bài 1.4 Viết chương trình nhận vào tháng.Xuất ra màn hình số ngày của tháng đó.

//Phụ lục

//Kiểm tra tháng nhập vào : Tháng hợp lệ > 0 && < 13

//Tháng có 31 ngày : 1 3 5 7 8 10 12

//Tháng có 30 ngày : 4 6 9 11

//Tháng 2 : Có 28 or 29 ngày (năm nhuận tháng 2 có 29)

//Năm nhuần còn là năm chia hết cho 400 hoặc chia hết cho 4 nhưng không chia hết cho 100

//Cú pháp của switch

//switch(expression)

//{

// case constant - expression:

// statement(s);

// break; //optional

// case constant - expression:

// statement(s);

// break; //optional

// ....

// default: //Optional

// statement(s);

//}

//Trong đó :

//expression: Là giá trị được truyền vào để so sánh với các điều kiện constant - expression.

//constant - expression : Là điều kiện để so sánh với giá trị của expression truyền vào.

//break : Là lệnh để thoát khỏi cấu trúc điều khiển switch case.

//default: Đây là giá trị mặc định, khi không có giá trị của expression nào thõa điều kiện constant - expression thì đoạn code bên trong default sẽ được thực thi.

#include<iostream>

using namespace std;

bool KiemTraNamNhuan(int nam);

int main4()

{

int n, nam;

cout << "CHUONG TRINH KIEM TRA SO NGAY CUA MOT THANG !\n";

cout << "Nhap thang va nam tai day de kiem tra : ";

cin >> n >> nam;

if (n > 0 && n < 13) // tháng hợp lệ từ 1 đến 12

{

if (nam <= 0)

{

cout << "Nhap nam khong hop le !\n";

/\*return 0;\*/

}

else // năm hợp lệ mới làm tiếp

{

switch (n)

{

case 1: case 3: case 5: case 7: case 8: case 10: case 12:

cout << " So ngay cua thang " << n << " la 31 ngay " << endl;

break;

case 4: case 6: case 9: case 11:

cout << "So ngay cua thang " << n << " la 30 ngay " << endl;

case 2:

if (KiemTraNamNhuan(nam) == 1) // true = 1 , năm nhuận

{

cout << "So ngay cua thang " << n << " la 29 ngay \n";

}

else // năm không nhuận

cout << "So ngay cua thang " << n << " la 28 ngay \n";

}

}

}

else

{

cout << "Nhap thang khong hop le !\n";

/\*return 0;\*/

}

system("pause");

return 0;

}

bool KiemTraNamNhuan(int nam)

{

if (nam % 400 == 0 || (nam % 4 == 0 && nam % 100 != 0))

{

return 1; // true = 1

}

else // Phần else có thể bỏ

return 0; // fales = 0

}

//Bài 1.5 Viết chương trình có chức năng như máy tính cầm tay(sử dụng switch)

//Các phép tính đơn giản + - \* /

//Hai số nguyên và 1 phép tính

#include<iostream>

#include<iomanip> // dùng fixed<<setprecision(2)

using namespace std;

int main5()

{

int n, a, b;

cout << "MENU LUA CHON CAC PHEP TINH\n";

cout << setfill('-');

cout << setw(60) << "-\n";

cout << "1. Phep tinh + \n";

cout << "2. Phep tinh - \n";

cout << "3. Phep tinh \* \n";

cout << "4. Phep tinh / \n";

cout << "5. Phep tinh % \n";

cout << setw(60) << "-\n";

cout << "Nhap phep tinh ban muon chon : ";

cin >> n;

switch (n)

{

case 1:

{

cout << "Phep tinh duoc lua chon la : Phep tinh +\n";

cout << "Nhap lan luot hai so can tinh (a,b) :";

cin >> a >> b;

cout << fixed << setprecision(2) << "a + b = " << a + b << endl;

break;

}

case 2:

{

cout << "Phep tinh duoc lua chon la : Phep tinh -\n";

cout << "Nhap lan luot hai so can tinh (a,b) :";

cin >> a >> b;

cout << fixed << setprecision(2) << "a - b = " << a - b << endl;

break;

}

case 3:

{

cout << "Phep tinh duoc lua chon la : Phep tinh \*\n";

cout << "Nhap lan luot hai so can tinh (a,b) :";

cin >> a >> b;

cout << fixed << setprecision(2) << "a \* b = " << a \* b << endl;

break;

}

case 4:

{

cout << "Phep tinh duoc lua chon la : Phep tinh / \n";

cout << "Nhap lan luot hai so can tinh (a,b) :";

cin >> a >> b;

if (b == 0)

cout << "Phep tinh loi chia cho 0\n";

else

cout << fixed << setprecision(2) << "a / b = " << a / b << endl;

break;

}

case 5:

{

cout << "Phep tinh duoc lua chon la: Phep tinh %\n";

cout << "Nhap lan luot hai so can tinh (a,b) :";

cin >> a >> b;

cout << fixed << setprecision(2) << "a % b = " << a % b << endl;

break;

}

default:

cout << "Nhap lua chon khong hop le !" << endl;

}

system("pause");

return 0;

}

//Bài 1.6 Viết chương trình nhận vào 3 số nguyên. Xuất ra màn hình 3 số đó được sắp xếp tăng dần

//Dùng hàm hoán vị nên khai báo thêm một ẩn phụ nữa

//Ẩn : a,b,c kiểu double

#include<iostream >

using namespace std;

int main6()

{

int a, b, c, tam;

cout << "Nhap lan luot cac so nguyen can so sanh : ";

cin >> a >> b >> c;

if (a > b)

{

tam = a;

a = b;

b = tam;

}

if (a > c)

{

tam = a;

a = c;

c = tam;

}

if (b > c)

{

tam = b;

b = c;

c = tam;

}

cout << "Cac so nguyen duoc sap xep tang dan la : " << a << " , " << b << " , " << c << endl;

system("pause");

return 0;

}

//Bài 1.7 Viết chương trình nhận vào số đo 3 cạnh của 1 tam giác. Kiểm tra xem 3 cạnh đó có hợp lệ hay không?

// Nếu hợp lệ thì tam giác đó là loại tam giác gì (đều, vuông,cân, vuông cân hay thường) ?

// Tam giác hợp lệ là : 1 cạnh < 2 cạnh còn lại cộng lại

// Tam giác đều : 3 cạnh bằng nhau

// Tam giác vuông : định lý pytago ( cạnh huyền bình phương = bình phương của hai cạnh gốc vuông kia cộng lại , thử 3 trường hợp

// Tam giác cân : hai cạnh bằng nhau, thử 3 trường hợp

// Tam giác vuông cân : điều kiện tam giác vuông + hai cạnh gốc vuông bằng nhau

// Tam giác thường

// Ẩn : a,b,c kiểu double

#include <iostream>

#include<iomanip>

using namespace std;

int main7()

{

double a, b, c;

cout << "CHUONG TRINH XAC DINH TAM GIAC " << endl;

cout << "Nhap vao ba canh cua tam giac " << endl;

cout << "Nhap vao canh thu nhat : "; cin >> a;

cout << "Nhap vao canh thu hai : "; cin >> b;

cout << "Nhap vao canh thu ba : "; cin >> c;

cout << setfill('-');// Chuyển từ mặc định setw" "

cout << setw(60) << "-" << endl;

if (a < (b + c) && b < (a + c) && c < (a + b)) // Điều kiện để tam giác hợp lệ

{

cout << "Tam giac hop le !\n";

if (a == b && b == c && c == a)

cout << "Kieu tam giac deu !\n";

else if (a == b || b == c || a == c)

cout << "Kieu tam giac can !\n";

else if (a \* a == b \* b + c \* c || b \* b == a \* a + c \* c || c \* c == a \* a + b \* b)

cout << "La kieu tam giac vuong !\n";

else if ((a \* a == b \* b + c \* c || b \* b == a \* a + c \* c || c \* c == a \* a + b \* b) && (a == b || b == c || a == c))

cout << "La kieu tam giac vuong can !\n";

else

cout << "La kieu tam giac thuong !\n";

}

else

cout << "Tam giac khong hop le !";

system("pause");

return 0;

}

//Bài 1.8 Viết chương trình nhận vào 1 số nguyên n(n > 0).

// Tính và xuất ra màn hình kết quả tổng các số từ 1 đến n.

// Kiểm tra đầu vào

// Ẩn : n,i,tong kiểu int

// Dùng while thì nhớ khai báo biến i trước

// Khai báo i = 0 hoặc 1 ... tùy trường hợp, biến tổng cũng vậy

// Có thể dùng while hoặc for

//Cú pháp while (không cần biết số lần lập như for)

//while (expression)

//statement;

//hoặc

//while (expression)

//{

// statements;

//}

//Cú pháp do while (Vòng lập do-while sẽ được thực thi ít nhất 1 lần.)

//do

//{

// statement;

//} while (condition);

//Cú pháp for (Phải biết trước số lần lập )

//for (init - statement; condition - expression; end - expression)

//{

// statements;

//}

//1. init - statement: phần này có mục đích định nghĩa và khởi tạo biến, chỉ được thực thi 1 lần duy nhất trong lần lặp đầu tiên.

//2. condition - expression : phần này gồm các biểu thức điều kiện, nếu biểu thức điều kiện đúng, các câu lệnh trong vòng lặp sẽ được thực thi.

//3. end - expression : phần này được thực thi cuối mỗi lần lặp, sau khi các câu lệnh trong vòng lặp for được thực thi.

//Phần này thường có mục đích tăng hoặc giảm giá trị các biến vòng lặp.Sau khi thực thi xong, vòng lặp quay lại kiểm tra điều kiện lặp ở bước 2.

#include<iostream>

using namespace std;

int main8()

{

int n;

do // Kiểm tra đầu vào

{

cout << "Nhap so nguyen tai day (n > 0) : ";

cin >> n;

if (n <= 0)

cout << "Nhap sai !";

} while (n <= 0);

//int i = 1; cách 1 thì khai báo biến i trước

int tong = 0;

// Cách 1

//while (i <= n)

//{

// tong += i; // tong = tong + i

// i++;

//}

// Cách 2

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

{

tong += i; // tong = tong + i

}

cout << "Tong cac so nguyen tu 1 den " << n << " la : " << tong << endl;

system("pause");

return 0;

}

//Bài 1.9 Viết chương trình nhận vào 1 số nguyên dương n > 1. Tính và xuất tích các số lẻ trong phạm vi từ 1 đến n.

// Kiểm tra đầu vào : n > 1

// Ẩn : int n, int i = 2 , int tong = 0

// Dùng for hoặc while

#include <iostream>

using namespace std;

int main9()

{

int n;

cout << "Nhap so nguyen n tai day : ";

cin >> n;

int tich = 1;

// Cách 1

int i = 1;

if (n > 1)

{

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

{

if (i % 2 == 1)

{

tich \*= i;

}

}\*/

while (i <= n)

{

tich \*= i; // tich = tich \* i;

i = i + 2;

}

}

else // n <= 1

{

cout << "Nhap so nguyen n khong hop le !";

}

cout << "Tong cac so nguyen le tu 1 den " << n << " la : " << tich << endl;

system("pause");

return 0;

}

//Bài 1.10 Tính và xuất ra màn hình kết quả tổng các số lẻ và tổng các số chẵn trong phạm vi từ 1 đến n

//Dùng for hoặc while đều được

//Kiểm tra đầu vào n

//Ẩn : int n, int i (nếu dùng while) , int tongle, int tongchan;

#include<iostream>

using namespace std;

int main10()

{

int n;

int tongle = 0, tongchan = 0;

do

{

cout << "Nhap so nguyen n tai day : ";

cin >> n;

if (n <= 1)

cout << "Nhap so nguyen n khong hop le !";

} while (n <= 1);

// Cách 1

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

//{

// if (i % 2 == 1) // lẻ

// {

// tongle += i;

// }

// else // chẵn

// {

// tongchan += i;

// }

//}

int i = 1;

while (i <= n)

{

if (i % 2 == 1) // lẻ

{

tongle += i;

}

else // chẵn

{

tongchan += i;

}

i++;

}

cout << "Tong cac so le tu 1 den " << n << " la : " << tongle << endl;

cout << "Tong cac so chan tu 1 den " << n << " la : " << tongchan << endl;

system("pause");

return 0;

}

//Bài 1.11 Viết chương trình nhận vào 1 số nguyên dương n.Xuất ra màn hình số chữ số và

// tổng các chữ số của n.Ví dụ : n = 3523 thì kết quả là n có 4 chữ số và tổng các chữ số là 13.

//Kiểm tra đầu vào : n > 0

//Ẩn : int n, int chuso , int tong

//Phương pháp tách :

// % 10 để lấy số cuối

// / (chia) cho 10 để bỏ đi số cuối

// Dùng vòng lập while tiện hơn

// Đúng

#include<iostream>

using namespace std;

int main11()

{

int n;

do

{

cout << "Nhap so nguyen n tai day : ";

cin >> n;

if (n <= 0)

cout << "Nhap so nguyen n khong hop le !\n";

} while (n <= 0);

int chuso = 0, tong = 0, dem = 0;

int tam = n;

while (tam > 0) // Nếu trong while là n thì n sau khi kết thúc vòng lập sẽ bị thay đổi

{

chuso = tam % 10;

tong += chuso; // tong = tong + chuso

dem++;

tam /= 10; // tam = tam / 10

}

//Cách 2 : dùng for

//for( tam ; tam > 0 ; tam /= 10 )

//{

// chuso = tam % 10;

// tong += chuso; // tong = tong + chuso

// dem++;

// }

cout << "So cac chu so cua " << n << " la : " << dem << endl;

cout << "Tong cac chu so cua " << n << " la : " << tong << endl;

system("pause");

return 0;

}

// Sai

//#include <iostream>

//using namespace std;

//int main()

//{

// int n, tong = 0;

// do

// {

// cout << "Nhap so nguyen duong tai day : ";

// cin >> n;

// if (n <= 0)

// cout << "Nhap sai nhap lai (n > 0)";

// } while (n <= 0);

// int soChuSo = 0;

//

// while (n > 0)

// {

// // vd n=345 ; n=34 ; n=3

// int chuso = n % 10; // chuso = 5 ; chuso = 4 ; chuso=3

// tong += chuso; // tong = 5 ; tong = 5 + 4 ; tong = 5 + 4 + 3

// soChuSo++; // =1 ;= 2 ;= 3

// n /= 10; // tam = 34 ; tam = 3 ; tam = 0 dung

// }

// cout << "Tong cac chu so cua " << n << " la : " << tong << endl;

// cout << "So cac chu so cua " << n << " la : " << soChuSo << endl;

// system("pause");

// return 0;

//}

// Nếu while (n > 0) thì biến n sau này bị đổi

//Bài 1.12 Viết chương trình nhận vào 1 số nguyên dương n.Xuất ra màn hình số đảo ngược của n.

// Ví dụ n là 352 thì số đảo ngược của nó là 253.

// Lưu ý : thực hiện việc kiểm tra giá trị n ban đầu nhập vào.Nếu n <= 0 thì thông

// báo nhập sai và cho phép nhập lại cho đến khi nào n > 0 như đề toán yêu cầu thì

// mới dừng việc nhập để tính toán.

//Dùng phương pháp tách

//Tách rồi nhân lên 1 hàng đơn vị rồi cộng số đã tách

//Dùng vòng lập while tiện hơn

//Ẩn : int n, int chuso, int sodao, tam

// Nhớ kiểm tra đầu vào, dùng do-while

#include<iostream>

using namespace std;

int main12()

{

int n;

do

{

cout << "Nhap so nguyen n tai day (n>0) : ";

cin >> n;

if (n <= 0)

cout << "Nhap so nguyen n khong hop le !\n";

} while (n <= 0);

int chuso;

int tam = n;

int sodao = 0;

while (tam > 0)

{

chuso = tam % 10;

sodao = sodao \* 10 + chuso;

tam /= 10;

}

/\*for (tam; tam > 0; tam /= 10)

{

chuso = tam % 10;

sodao = sodao \* 10 + chuso;

}\*/

cout << "So dao nguoc cua " << n << " la : " << sodao << endl;

system("pause");

return 0;

}

//Bài 1.13 Viết chương trình nhận vào 1 số nguyên n. Xuất ra màn hình thông báo n có phải là số nguyên tố hay không ?

// Biết rằng số nguyên tố là số từ 2 trở lên, chỉ chia hết cho 1 và cho chính bản thân nó.

//Bài này không cần kiểm tra đầu vào

//Có thể dùng hàm bool nếu thích

//Dùng cờ

//Dùng for hoặc while để lấy n chia cho i , điều kiện i <= sqrt (n)

//Dùng while tiện hơn (vì không biết trước số lần lập)

//Cú pháp ép kiểu

// static\_cast<kiểu int hay ...>(số cần ép kiểu)

//Hàm bool thì

// True là đúng , 1 cũng là đúng

// False là sai , 0 cũng là sai

#include <iostream>

#include<iomanip> // dùng setfill('-') và setw(60)<<"-";

using namespace std;

int main13()

{

int n;

cout << "CHUONG TRINH KIEM TRA SO NGUYEN TO \n";

cout << setfill('-'); // chuyển từ mặc định setw " " sang setw "-"

cout << setw(60) << "-" << endl;

cout << "Nhap so can kiem tra : ";

cin >> n;

bool songuyento = 1;

int i = 2;

if (n >= 2)

{

while (i <= sqrt(static\_cast<double>(n)) && songuyento == 1) // Ép kiểu n về double để căn bậc 2 của nó chính xác

{ // Chỉ cần chia hết cho 1 số i thì n không cón là số nguyên tố

if (n % i == 0)

songuyento = 0;

i++;

}

}

else // n < 2

songuyento = 0;

if (songuyento == 1)

cout << "So nhap vao la so nguyen to !\n";

else // songuyento == 0

cout << "So nhap vao khong la so nguyen to !\n ";

system("pause");

return 0;

}

//Bài 1.13 (NÂNG CAO) Viết chương trình nhận vào 2 số nguyên dương n1 và n2(n1 < n2).

// Đếm và xuất kết quả xem từ n1 đến n2 có bao nhiêu số là số nguyên tố ?

//Kiểm tra số nhập vào n1 < n2

//Vòng chạy của while hoặc for thì <= sqrt (n2)

//Nhớ ép kiểu n2 về double để vòng chạy đúng hơn (căn bậc 2 của n2)

//Ẩn : int n1, int n2, dem

//Bài này có phần đếm (có bao nhiêu số nguyên tố) nên dùng thêm hàm tách biệt để nó dễ hơn

//Dùng hầm nên không cần dùng cờ

#include<iostream>

using namespace std;

bool kiemtrasonguyento(int n);

int main13n()

{

int n1, n2;

do

{

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

cin >> n1;

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

cin >> n2;

if (n1 <= 0 || n1 > n2)

cout << "Nhap so nguyen duong khong hop le !\n";

} while (n1 <= 0 || n1 >= n2);

int dem = 0;

cout << "Cac so nguyen to trong khoang tu " << n1 << " den " << n2 << " la : ";

for (int i = n1; i <= n2; i++)

{

if (kiemtrasonguyento(i) == 1)

{

cout << i << " ";

dem++;

}

}

//Cách 2

/\*int i = n1;

while (i <= n2)

{

if (kiemtrasonguyento(i) == 1)

{

cout << i << " ";

dem++;

}

i++;

}\*/

cout << endl;

cout << "So cac so nguyen to tu " << n1 << " den " << n2 << " la : " << dem << endl;

system("pause");

return 0;

}

bool kiemtrasonguyento(int n)

{

if (n < 2)

return 0; //false

int i = 2;

while (i <= sqrt(static\_cast<double>(n)))

{

if (n % i == 0)

return 0;

i++;

}

//Cách 2

/\*for (int i = 2; i <= sqrt(static\_cast<double>(n)); i++)

{

if (n % i == 0)

return false;

}\*/

return 1; // true

}

//Bài 1.14 Viết chương trình nhận vào 1 số nguyên n > 0. Tính và xuất tổng các chữ số lẻ của n.

//Vd : n= 12345 thì tong=9

//Kiểm tra đầu vào

//Dùng for hay while đều được

//Dùng phương pháp tách

#include<iostream>

using namespace std;

int main14()

{

int n;

do

{

cout << "Nhap so nguyen n tai day (n>0) : ";

cin >> n;

if (n <= 0)

cout << "Nhap so nguyen n khong hop le !\n";

} while (n <= 0);

int chuso;

int tam = n;

int sum = 0;

while (tam > 0)

{

chuso = tam % 10;

if (chuso % 2 == 1) // tìm số lẻ

sum += chuso;//sum = sum + chuso

tam /= 10;

}

cout << "Tong cac chu so le cua " << n << " la : " << sum << endl;

system("pause");

return 0;

}

//Bài 1.15 Viết chương trình nhận vào 2 số nguyên dương n1 và n2(n1 < n2).

// Đếm và xuất ra màn hình kết quả có bao nhiêu số là ước của 10 trong phạm vi từ n1 đến n2.

//Dùng vòng lập while or for

//Kiểm tra đầu vào : n1 > 0, n1 < n2 , n2 > 0

//Ẩn : int n1, int n2, int dem ;

//Ước của 10 là : 1,2,5,10

//Cẩn thận bị vòng lập vô tận, bị là còn 000 điểm luôn

#include<iostream>

using namespace std;

int main15()

{

int n1, n2;

do

{

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

cin >> n1;

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

cin >> n2;

if (n1 < 0 || n2 < 0 || n1 >= n2)

cout << "Nhap so nguyen khong hop le !\n";

} while (n1 <= 0 || n2 <= 0 || n1 >= n2);

int i = n1;

int dem = 0;

cout << "Uoc cua so 10 trong pham vi tu " << n1 << " den " << n2 << " la : ";

while (i <= n2)

{

if (10 % i == 0) //Tìm ước của 10 (1,2,5,10)

{

cout << i << " ";

dem++;

}

i++;

}

cout << endl;

cout << "So luong uoc cua so 10 trong pham vi tu " << n1 << " den " << n2 << " la : " << dem << endl;;

system("pause");

return 0;

}

//Bài 1.15 (NÂNG CAO) Viết chương trình nhận vào 2 số nguyên dương n1 và n2(n1 < n2).

//Đếm xem từ n1 đến n2 có bao nhiêu số chẵn ? bao nhiêu số lẻ ? bao nhiêu số là ước của 10 ?

//Kiểm tra đầu vào : n1 > 0, n2 > 0 , n1 < n2

//Ẩn : int n1, int n2, int demle, int demchan, int demuoc

//Dùng vòng lập coi chừng bị lập vô tận

#include<iostream>

#include<iomanip> // Dùng setfill('-') , setw(60) << "-"

using namespace std;

int main15n()

{

int n1, n2;

do {

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

cin >> n1;

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

cin >> n2;

if (n1 < 0 || n2 < 0 || n1 >= n2)

cout << "Nhap so nguyen khong hop le !\n";

} while (n1 <= 0 || n2 <= 0 || n1 >= n2);

cout << setfill('-');

cout << setw(60) << "-" << endl;

int i = n1;

int demle = 0, demchan = 0, demuoc = 0;

while (i <= n2)

{

if (i % 2 == 0) // số chẵn

demchan++;

if (i % 2 == 1) // số lẻ

demle++;

if (10 % i == 0) // ước của 10

demuoc++;

i++; //Không có là bị lập vô tận

}

cout << "So cac so chan trong khoang tu " << n1 << " den " << n2 << " la : " << demchan << endl;

cout << "So cac so le trong khoang tu " << n1 << " den " << n2 << " la : " << demle << endl;

cout << "So cac uoc cua so 10 trong khoang tu " << n1 << " den " << n2 << " la : " << demuoc << endl;

system("pause");

return 0;

}

//Bài 1.16 Viết chương trình nhận vào 1 số nguyên n > 0. Xuất ra màn hình các giá trị thừa

// số nguyên tố của n.Ví dụ n = 126 thì các thừa số nguyên tố là 2, 3, 3, 7.

//Dùng 2 vòng lập : vòng 1 cho i chạy tới n, vòng 2 cho n chia cho i

//Vòng lập 1 bao luôn vòng lập thứ 2

//Sử dụng biến đếm để lát xuất số cho đẹp, dem = 1 xuất bình thường, đếm > 1 xuất kèm "^", n > i thì xuất thêm "\*"

//Kiểm tra đầu vào

#include<iostream>

using namespace std;

int main16()

{

int n;

do

{

cout << "Nhap so nguyen n tai day (n > 0): ";

cin >> n;

if (n <= 0)

cout << "Nhap so nguyen n khong hop le ! ";

} while (n <= 0);

int i = 2; // Vì số nào cũng chia hết cho 1

int dem;

cout << "So nguyen " << n << " = ";

while (i <= n)

{

dem = 0; // mỗi vòng lập lớn thì dem lại trở về 0

while (n % i == 0)

{

dem++;

n /= i; // n= n/i (tách thừa số nguyên tố ra từ từ)

}

if (dem) // Nếu có biến đếm ( dem > 0)

{

cout << i;

if (dem > 1) // Đếm >= 2

cout << "^" << dem;

if (n > i) // n > i thôi vì khi n=i thì chỉ còn 1 thừa số nguyên tố cuối nên không cần hiện dấu \*

cout << "\*";

}

i++; // Tăng i để vòng lập không bị vô hạn

}

cout << endl;

system("pause");

return 0;

}

//Bài 1.17 Viết chương trình nhận vào 1 số ở dạng nhị phân.Xuất ra màn hình kết quả số đó chuyển qua hệ thập phân.

//Kiểm tra đầu vào : số nhị phân là bao gồm các số (0,1), số lớn hơn 0 DÙNG HÀM PHỤ ĐỂ KIỂM TRA

// 543210

//VD: Hệ 2 = 111110

// Hệ 10 = 0\*2^0 + 1\*2^1 + 1\*2^2 + 1\*2^3 + 1\*2^4 + 1\*2^5

//Dùng phương pháp tách

//Dùng for or while

#include<iostream>

#include<cmath> //dùng hàm pow(cơ số, số mũ)

using namespace std;

bool ktsonhiphan(int n);

int main17()

{

int n;

do

{

cout << "Nhap so nhi phan tai day : ";

cin >> n;

if (ktsonhiphan(n) == 0) // =0 là false

cout << "Nhap so nhi phan khong hop le !";

} while (ktsonhiphan(n) == 0); // =0 là false

int chuso, sothapphan = 0, mu = 0;

int tam = n; // gán tam = n để lát cout n không bị thay đổi

while (tam > 0)

{

chuso = tam % 10;

sothapphan += chuso \* pow(2, mu);

tam /= 10; // tam = tam / 10

mu++;

}

cout << "So nhi phan " << n << " = " << sothapphan << " so thap phan " << endl;

system("pause");

return 0;

}

bool ktsonhiphan(int n)

{

if (n > 0)

{

while (n % 10 > 1)

return 0; //false

}

else //

return 0; //false

return 1; // true

}

//Bài 1.18 Viết chương trình nhận vào 1 số nguyên n > 0. Tách từng chữ số của n và thể hiện chúng dưới dạng cách nhau một khoảng cách.

// Ví dụ n = 3456 thì kết quả từng chữ số là 3 4 5 6

// Dùng hai phương pháp liên tiếp

// Tách lần thứ nhất rồi nhân lên cho cho 10 sau mỗi đợt tách

// Tách lần thứ hai là xong

#include<iostream>

using namespace std;

int main18()

{

int n;

do

{

cout << "Nhap so nguyen n tai day (n > 0) : ";

cin >> n;

if (n <= 0)

cout << "Nhap so nguyen n khong hop le !\n";

} while (n <= 0);

int tam = n;

int chuso;

int sodao = 0;

while (tam > 0) // 3456

{

chuso = tam % 10; // 6 ; 5 ; 4 ; 3

sodao = sodao \* 10 + chuso; // 6 ; 65 ; 654 ; 6543

tam /= 10; // 345 ; 34 ; 3

}

int sotach;

cout << "So duoc tach ra la : ";

while (sodao > 0)

{

sotach = sodao % 10;

cout << sotach << " ";

sodao /= 10;

}

cout << endl;

system("pause");

return 0;

}

// Cách phức tạp hơn

#include <iostream>

using namespace std;

int main18n()

{

int n, sochia = 1, dem = 0;

cout << "Nhap so nguyen (n > 0) : ";

cin >> n;

int tam = n;

if (n > 1)

{

while (tam > 10) //3456

{

tam = tam / 10; //345 ; 34 ; 3

sochia = sochia \* 10; //10 ; 100 ; 1000

}

tam = n; // 3456

while (tam > 0)

{

dem = tam / sochia; // 3456/1000=3

tam = tam % sochia; // 3456%1000=456

sochia = sochia / 10; //1000/10=100

cout << dem << " ";

}

}

else

cout << " Nhap sai ! " << endl;

system("pause");

return 0;

}

//Bài 1.19 Viết chương trình cho người dùng nhập giá trị cho m số nguyên. Tính và xuất

//tổng các số vừa nhập.Giao diện chương trình khi thực hiện được mô phỏng ở ví dụ sau :

//Ban can bao nhieu so : 3

//Nhap so thu 1 : 6

//Nhap so thu 2 : 4

//Nhap so thu 3 : 2

//Tong 3 so vua nhap la : 12

//Dùng 1 vòng lập

//Vòng lập thứ nhất để nhập số cho i và tính tổng luôn

//Dùng for hay while đều được

//Nhớ kiểm tra đầu vào và đừng cho lập vô hạn

#include<iostream>

using namespace std;

int main19()

{

int n, m;

int tong = 0;

do

{

cout << "Ban can nhap bao nhieu so (n > 0) : ";

cin >> n;

if (n <= 0)

cout << "Nhap so khong hop le !\n";

} while (n <= 0);

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

{

cout << "Nhap so thu " << i << " : ";

cin >> m; // không dùng cin >> i được

tong += i;

}

cout << "Tong " << n << " vua nhap la : " << tong << endl;

system("pause");

return 0;

}

//Bài 1.20 Viết chương trình nhận vào 1 số nguyên dương n. Tính và xuất kết quả tính n giai thừa.

// 0!= 1

// n!=1\*2\*3\*...\*n

#include<iostream>

using namespace std;

int main20()

{

int n;

do

{

cout << "Nhap so can tim giai thua : ";

cin >> n;

if (n < 0)

cout << "Nhap sai ! ";

} while (n < 0);

int giaithua = 1;

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

{

giaithua \*= i; // giathua=giaithua\*i;

}

cout << "Giai thua cua " << n << " la : " << "( " << n << "! ) = " << giaithua << endl;

system("pause");

return 0;

}

//Bài 1.21 Viết chương trình nhận vào 2 số nguyên dương là chiều dài d và chiều rộng r của hình chữ nhật.

// Xuất ra hình chữ nhật dưới dạng các dấu \* .

// Ví dụ d = 3, r = 5 thì kết quả :

// \*\*\*\*\*

// \*\*\*\*\*

// \*\*\*\*\*

// Kiểm tra đầu vào: dài và rộng > 0

// Dùng 2 vòng lập : vòng lập thứ 1 bao luôn vòng lập thứ 2

// Vòng lập 1 chạy chiều dài ( từ trên xuống)

// Vòng lập 2 chạy chiều rộng ( từ trái sang phải)

// Dùng for dễ kiểm soát để nó không bị lập vô tận

#include<iostream>

using namespace std;

int main21()

{

int n, m;

do

{

cout << "Nhap chieu dai hinh chu nhat : ";

cin >> n;

cout << "Nhap chieu rong hinh chu nhat : ";

cin >> m;

if (n <= 0 || m <= 0)

cout << "Nhap sai ! ";

} while (n <= 0 || m <= 0);

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

{

for (int i = 1; i <= m; i++) // mỗi vòng lập một định nghĩa nên int i trùng không sao

{

cout << "\*";

}

cout << endl; // xuống dòng sau mỗi lần tăng chièu dài

}

/\*cout << endl;\*/

system("pause");

return 0;

}

//Bài 1.22 Viết chương trình nhận vào chiều cao h của một hình tam giác(h > 0).

// Xuất ra màn hình tam giác có chiều cao h dưới dạng các dấu\* .

// Ví dụ h = 3 thì kết quả :

// \*

// \*\*

// \*\*\*

//Kiểm tra đầu vào

//Dùng 2 vòng lập lồng vào nhau

//Vòng lập 1 chạy chiều dài (từ trên xuống dưới) : i<= h

//Vòng lập 2 chạy chiều rộng (từ trái sang phải): int k <= i

#include<iostream>

using namespace std;

int main22()

{

int n;

do

{

cout << "Nhap chieu cao tam giac : ";

cin >> n;

} while (n <= 0);

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

{

for (int k = 1; k <= i; k++) // int k cho khác với i, có điều kiện k <= i

{

cout << "\*" << " ";

}

cout << endl;

}

system("pause");

return 0;

}

//Bài 1.23 Viết chương trình nhận vào 1 số nguyên dương n(n > 0).

// Tính và xuất tổng các số lẻ từ 1 đến n.Ví dụ : n = 4 thì kết quả là 4.

//Kiểm tra đầu vào n > 0

//Dùng vòng lập

//Ẩn : int n, int tongle

#include<iostream>

using namespace std;

int main23()

{

int n, tongle = 0;

do

{

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

cin >> n;

if (n <= 0)

cout << "Nhap so nguyen n khong hop le !\n";

} while (n <= 0);

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

{

if (i % 2 == 1)

tongle += i;

}

cout << "Tong cac so le tu 1 den " << n << " la : " << tongle << endl;

system("pause");

return 0;

}

//Bài 1.24 Viết chương trình nhận vào 1 số nguyên dương n(n > 0).

// Xuất ra màn hình bảng cửu chương ứng với số nguyên n.Ví dụ n = 3 thì kết quả xuất ra màn hình :

// 3 x 1 = 3

// 3 x 2 = 6

// …

// 3 x 10 = 30

//Dùng vòng lập chạy từ 1 đến 10

//Kiểm tra đầu vào

//Ẩn : int n

#include<iostream>

#include<iomanip> //dùng setfill và setw

using namespace std;

int main24()

{

int n;

cout << "CHUONG TRINH IN BANG CUU CHUONG \n";

cout << setfill('-');

cout << setw(60) << "-" << endl;

do

{

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

cin >> n;

if (n <= 0)

cout << "Nhap so nguyen khong hop le !\n";

} while (n <= 0);

for (int i = 1; i <= 10; i++)

{

cout << n << "x" << i << "=" << n \* i << endl;

}

system("pause");

return 0;

}

//Bài 1.25 Viết chương trình nhận vào 1 số nguyên dương c là chiều dài cạnh của hình

// vuông(c > 0). Xuất ra hình vuông dưới dạng các dấu \* .

// a.Hình vuông với n = 4

// \*\*\*\*

// \*\*\*\*

// \*\*\*\*

// \*\*\*\*

// b.Hình vuông với n = 4

// \*\*\*\*

// \* \*

// \* \*

// \*\*\*\*

//Câu a:

//Dùng 2 vòng lập lồng nhau, int i giống nhau cũng được

//Dùng for hay while đều được, cẩn thận bị lập vô tận

//Dùng hàm cho câu a,b

//Câu b hơi khó xíu:

//Dùng 2 vòng lập lồng nhau

//Vòng 1 chạy chiều dài (trên xuống dưới) : i<=n

//Vòng 2 chạy chiều rộng (trái sang phải), dùng if kết hợp i<=n hoặc k<=n

//Câu a và b dùng hàm void (vì không cần trả trị như tính toán), dùng switch để lựa chọn câu muốn xuất hiện

#include<iostream>

#include<iomanip>

using namespace std;

void cauA(int n);

void cauB(int n);

int main25()

{

int n;

cout << "LUA CHON HINH XUAT HIEN \n";

cout << " 1. Hinh vuong dac \n";

cout << " 2. HInh vuong rong\n";

cout << setfill('-');

cout << setw(60) << "-" << endl;

cout << "Ban muon chon hinh : ";

int m;

cin >> m;

do

{

cout << "Nhap so nguyen duong n ( n > 0) : ";

cin >> n;

if (n <= 0)

cout << "Nhap so nguyen duong n khong hop le !\n";

} while (n <= 0);

switch (m)

{

case 1:

cauA(n);

break;

case 2:

cauB(n);

break;

default:

cout << "Nhap lua chon khong hop le !\n";

}

system("pause");

return 0;

}

void cauA(int n)

{

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

{

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

{

cout << "\*";

}

}

}

void cauB(int n)

{

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

{

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

{

if (i == 1 || i == n)

cout << "\*";

else

{

if (k == 1 || k == n)

cout << "\*";

else

cout << " ";

}

}

cout << endl;

}

}

//Bài 1.26 Viết chương trình nhận vào chiều cao h của một hình tam giác(h > 0).Xuất ra

// màn hình tam giác có chiều cao h dưới dạng các dấu\* .

// a.Tam giác với h = 5: i==k; i==n ; k==1

// \*

// \*\*

// \* \*

// \* \*

// \*\*\*\*\*

// a2.Tam giac với h = 5:

// \*

// \*\*

// \*\*\*

// \*\*\*\*

// \*\*\*\*\*

// b.Tam giác với h = 4

// \*

// \*\*

// \*\*\*

// \*\*\*\*

// b2.Tam giác với h = 4 : k==n ; i==n ; k>n-i && k < n+1

// \*

// \*\*

// \* \*

// \* \*

// \*\*\*\*\*

// c.Tam giác với h = 4 : i=1; k <= n-i

// \*\*\*\*

// \*\*\*

// \*\*

// \*

// d.Tam giác với h = 4

// \*\*\*\*

// \*\*\*

// \*\*

// \*

// e.Tam giác với h = 4

// \*

// \* \*

// \* \* \*

// \* \* \* \*

// f.Tam giác với h = 4

// \* \* \* \*

// \* \* \*

// \* \*

// \*

//Kiểm tra đầu vào h>0

//Dùng hàm void cho từng câu (do không cần trả trị)

//Dùng switch để lựa chọn hình xuất ra

//Câu a dùng 2 vòng lập:

//Vòng lặp 1 chiều dài (trên xuống): i<=n

//Vòng lập 2 chiều rộng (trái sang phải): k<=i

//Điều kiện để xuất hiện hình tam giác rỗng là : i==k; k==1; i == n

//Câu a2 dùng 2 vòng lập

//Vòng lập 1 chiều dài (trên xuống): i<=n

//Vòng lập 2 chiều rồng (trái sang phải) : k<=i

//Điều kiện xuất hiện hình :

// k<=i xuất "\*"

//Câu b dùng 2 vòng lập

//Vòng lập 1 chiều dài (trên xuống ) : i<=n

//Vòng lập 2 chiều rộng (trái sang phải ) : k <= n

//Điều kiện để xuất hiện tam giác đặc :

// k<(n-i) xuất " "

// else xuất "\*"

//Câu b2 dùng 2 vòng lập

//Vòng lập 1 chiều dài (trên xuống): i<= n;

//Vòng lập 2 chiều rộng (trái sang): k<=n;

//Điều kiện xuất hiện vòng lập :

// k==n; i== n xuất "\*"

// if (k<= n-i) xuất " "

//Câu c dùng 2 vòng lập lồng nhau

//Vòng lập 1 chiều dài (trên xuống): int i = n; i >= 1; i++

//Vòng lập 2 chiều rộng (trái sang): int k=1; k<=i; k++

//Câu d dùng 3 vòng lập, 1 vòng lập lòng 2 vòng lập chạy song song

//Vòng lập 1 chạy chiều dài (trên xuống) : int i = 1; i <= n; i++

//Vòng lập 2 3 chạy chiều rộng (trái sang phải) :

//Vòng lập 2 chạy " " : int k= 0; k < i; k++

//Vòng lập 3 chạy "\*" : int q = i ; q <= n ; q++

//Câu e dùng 3 vòng lập, 1 vòng lập lòng 2 vòng lập chạy song song

//Vòng lập q chạy chiều dài (trên xuống): int i ; i<= n; i++

//Vòng lập 2,3 chạy chiều rộng (trái sang phải)

//Vòng lập 2 chạy " ": int k = i ; k <n; k++

//Vòng lập 3 chạy "\*": int q = 1; q <=(2\*i-1); q++

//Câu f

//.... KHÓ QUÁ BỎ QUA

#include<iostream>

using namespace std;

void tamgiacrongA(int n);

void tamgiacdacA2(int n);

void tamgiacdacB(int n);

//void tamgiacrongB2(int n);

void tamgiacdacC(int n);

void tamgiacdacD(int n);

void tamgiacdacE(int n);

void tamgiacdacF(int n);

int main26()

{

int n;

do

{

cout << "Nhap so nguyen duong n (n>0) : ";

cin >> n;

if (n <= 0)

cout << "Nhap so n khong hop le !\n";

} while (n <= 0);

tamgiacrongA(n);

cout << endl;

tamgiacdacA2(n);

cout << endl;

tamgiacdacB(n);

cout << endl;

/\*tamgiacrongB2(n);\*/

tamgiacdacC(n);

cout << endl;

tamgiacdacD(n);

cout << endl;

tamgiacdacE(n);

cout << endl;

tamgiacdacF(n);

system("pause");

return 0;

}

void tamgiacrongA(int n)

{

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

{

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

{

if (i == k || i == n || k == 1)

cout << "\*";

else

cout << " ";

}

cout << endl;

}

}

void tamgiacdacB(int n)

{

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

{

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

{

if (k <= (n - i))

cout << " ";

else

cout << "\*";

}

cout << endl;

}

}

void tamgiacdacA2(int n)

{

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

{

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

cout << "\*";

cout << endl;

}

}

//void tamgiacrongB2(int n)

//{

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

// {

// for (int k = 1; k <= n; k++)

// {

// if (k==n || i==n || (k > (n-i) && k < (n + i) ))

// cout <<"\*";

// }

// cout << endl;

// }

//}

void tamgiacdacC(int n)

{

for (int i = n; i >= 1; i--)

{

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

cout << "\*";

cout << endl;

}

}

void tamgiacdacD(int n)

{

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

{

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

cout << " ";

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

cout << "\*";

cout << endl;

}

}

void tamgiacdacE(int n)

{

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

{

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

cout << " ";

for (int q = 1; q <= (2 \* i - 1); q++)

cout << "\*";

cout << endl;

}

}

void tamgiacdacF(int n)

{

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

{

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

cout << " ";

for (int q = 1; q <= (2 \* n - (2 \* i - 1)); q++)

cout << "\*";

cout << endl;

}

}

//Bài 1.27 Viết các hàm sau đây và chương trình kiểm tra các hàm đã viết :

// a.Hàm tính diện tích hình chữ nhật.

// b.Hàm tính chu vi hình chữ nhật

#include<iostream>

using namespace std;

double dtHCN(double n, double m);

double cvHCN(double n, double m);

int main27()

{

double n, m;

cout << "Nhập chieu dài HCN : ";

cin >> n;

cout << "Nhap chieu rong HCN : ";

cin >> m;

cout << "Dien tich HCN la : " << dtHCN(n, m);

cout << endl;

cout << "Chu vi HCN la : " << cvHCN(n, m);

cout << endl;

system("pause");

return 0;

}

double dtHCN(double n, double m)

{

return n \* m;

}

double cvHCN(double n, double m)

{

return (n + m) \* 2;

}

//Bài 1.28 Viết các hàm sau đây và chương trình kiểm tra các hàm đã viết :

// a.Hàm tìm số lớn nhất giữa 2 số nguyên.

// b.Hàm tìm số lớn nhất giữa 3 số nguyên.

// Xây dựng hàm tìm số lớn nhất của hai số nguyên.

// Toán tử 3 ngôi

// Cú pháp

// condition ? true\_value : false\_value;

// vd : if ( a >= b ? a : b )

#include<iostream>

using namespace std;

int soSanh2so(int a, int b);

int soSanh3so(int a, int b, int c);

int main28()

{

int a, b, c;

cout << "Nhap so nguyen thu 1 : ";

cin >> a;

cout << "Nhap so nguy thu 2 : ";

cin >> b;

cout << "Nhap so nguyen thu 3 : ";

cin >> c;

cout << "So lon nhat trong 2 so nguyen dau la : " << soSanh2so(a, b);

cout << endl;

cout << "So lon nhat trong 3 so nguyen la : " << soSanh3so(a, b, c) << endl;

system("pause");

return 0;

}

int soSanh2so(int a, int b)

{

int max = a;

if (max < b)

max = b;

return max;

// Cách 2 dùng toán tử 3 ngôi

// return ( a >= b ? a : b) ;

}

int soSanh3so(int a, int b, int c)

{

int max = a;

if (max < b)

max = b;

if (max < c)

max = c;

return max;

}

//Bài 1.29 Viết hàm kiểm tra một số nguyên n có phải là số hoàn thiện hay không ? Viết chương

// trình kiểm tra hàm vừa viết.Biết rằng số hoàn thiện là số có tổng các ước số(kể cả 1)

// bằng chính nó.Ví dụ : 6 là số hoàn thiện vì 6 = 1 + 2 + 3 (1, 2, 3 là các ước của 6).

//Số honaf thiện là số mà tổng các ước của nó bằng chính nó

#include<iostream>

using namespace std;

bool KTsohoanthien(int n);

int main29()

{

int n;

do

{

cout << "Nhap so nguyen n : ";

cin >> n;

if (n <= 0)

cout << "Nhap so nguyen khong hop le ! \n";

} while (n <= 0);

if (KTsohoanthien(n) == 1)//true

cout << n << " la so hoan thien \n";

else //

cout << n << " khong la so hoan thien\n";

system("pause");

return 0;

}

bool KTsohoanthien(int n)

{

int tong = 0;

for (int i = 1; i <= n / 2; i++) //i chạy từ 1 tới n/2 vì hấu hết các số chỉ chia cho số < 1/2 của nó

{

if (n % i == 0)

tong += i;

if (tong == n)

return 1; // return true

//không cần else trong trường hợp này (có cũng được không cũng không sau)

}

}

//Bài 1.30 Viết hàm kiểm tra một số nguyên n có phải là số đối xứng(Palindrome) hay không ?

// Viết chương trình kiểm tra hàm vừa viết.Biết rằng số đối xứng là số có số đảo ngược

// của nó bằng chính nó.

#include<iostream>

using namespace std;

bool check(int n);

int main30()

{

int n;

do

{

cout << "Nhap so nguyen n : "; cin >> n;

if (n < 0)

cout << "Nhap so nguyen khong hop le !\n";

} while (n < 0);

if (check(n) == 1)//true

cout << n << " la so dao nguoc !\n";

else

cout << n << " khong la so dao nguoc !\n";

system("pause");

return 0;

}

bool check(int n)

{

int sodau = n, sodu, sosau = 0;

while (n > 0)

{

sodu = n % 10;

sosau = sosau \* 10 + sodu;

n /= 10; //Điều kiện vòng lập (sau mỗi lần lập n giảm đi 10 lần)

}

if (sosau == sodau)

return 1; //return true

//Không cần else

}

//Bài 1.31 Viết các hàm sau đây và chương trình kiểm tra các hàm đã viết :

// a.Hàm tính n!(n là số nguyên không âm).

// b.Hàm tính tổ hợp n chập k theo công thức

// c.Hàm in tam giác Pascal có chiều cao h

// Giai thừa số lớn nên dùng kiểu long long

// 0!= 1

// n!=1\*2\*3\*...\*n

// b/ Tổ hợp n chập k : n! / (k!\*(n-k)!)

#include<iostream>

#include<iomanip> //dùng setw

using namespace std;

long long gt(int n);

long long tohop(int n, int k);

int main31()

{/\*

int n;

int k;

do

{

cout << "Nhap so nguyen n (n >0) : "; cin >> n;

if (n < 0)

cout << "Nhap sai nhap lai !\n";

} while (n < 0);

cout << "Giai thua cua so nguyen " << n << " la : " << giaithua(n);

cout << "To hop cua " << n << " chap " << k << " la : " << tohop(n, k);\*/

int h;

cout << "Nhap do cao Pascal : "; cin >> h;

cout << " K | " << setw(6);

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

cout << i << setw(8);

cout << endl;

cout << setw(1);

for (int i = 0; i < 2 \* h; i++)

cout << "-" << setw(4);

cout << endl;

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

{

cout << setw(3) << i << " | ";

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

{

cout << setw(8) << tohop(j, i);

}

cout << endl;

}

system("pause");

return 0;

}

long long gt(int n)

{

long long kq = 1;

if (n == 0)

return 1;

else

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

{

kq \*= i;

} return kq;

}

long long tohop(int k, int n)

{

return gt(n) / (gt(k) \* gt(n - k));

}

//Bài 8.1 Viết chương trình thực hiện các công việc sau đây :

// a) Khai báo một mảng alpha lưu 50 phần tử kiểu double.

// b) Gán giá trị của 25 phần tử đầu tiên là bình phương của chỉ số phần tử, 25 phần

// tử còn lại có giá trị là lập phương của chỉ số.

//2 Cách định nghĩa

//Cách 1 : #define Tên \_\_ Số (Tên thường viết hoa)

// #define MAX 50

//Cách 2 : const Kiểu dữ liệu \_\_ Tên \_\_ = Số

// const int MAX = 50

//Bình phương là mũ 2

//Lập phương là mũ 3

//Khai báo 1 chiều

//Kiểu dữ liệu \_\_ Tên[số đã định nghĩa]

//#difine MAX 50

//double arr[MAX];

#include<iostream>

using namespace std;

int main8\_1()

{

const int MAX = 50;

double alpha[MAX];

for (int i = 0; i <= 49; i++) //Mảng nên chạy từ 0 đến < n

{

if (i < 25)

alpha[i] = pow(i, 2);

else // >= 25

alpha[i] = pow(i, 3);

}

for (int i = 0; i <= 49; i++)

{

cout << alpha[i] << " ";

if ((i + 1) % 10 == 0) // Số chia hết cho 10 thì xuống hàng vd(10,20,...)

cout << endl;

}

system("pause");

return 0;

}

//Bài 8.2 Viết chương trình thực hiện các công việc sau đây :

// a) Định nghĩa hằng MAXSIZE là 100.

// b) Khai báo mảng values có thể lưu số phần tử tối đa là MAXSIZE, kiểu của mỗi

// phần tử là double.

// c) Khai báo currentSize lưu số phần tử hiện có trong mảng.

// d) Nhập vào giá trị của currentSize.

// e) Nhập vào các giá trị trong mảng values, số phần tử đọc vào là currentSize.

// f) In các phần tử trong mảng values trên một dòng, cách nhau bởi dấu phẩy.

#include<iostream>

using namespace std;

int main8\_2()

{

const int MAXSIZE = 100;

double values[MAXSIZE];

int currentSize;

cout << "Nhap so luong phan tu cua mang: ";

cin >> currentSize;

for (int i = 0; i < currentSize; i++) //Vòng lập nhập các số vào mảng

{

cin >> values[i];

}

for (int i = 0; i < currentSize; i++) //Vòng lập xuất mảng đã nhập

{

// Chia ra 2 trường hợp để xuất

// Chưa cuối xuất kèm dấu " , "

// Cuối xuất không kèm dấu

if (i == currentSize - 1)// i = phan tu cuoi cung, xuong hang

{

cout << values[i] << endl;;

break;

}

else

cout << values[i] << " , ";

}

system("pause");

return 0;

}

//Bài 8.3 Viết chương trình nhập vào một mảng số nguyên(tối đa 100 phần tử), tìm và in ra

//phần tử lớn nhất của mảng.

#include<iostream>

using namespace std;

int main8\_3()

{

const int MAXSIZE = 100;

double values[MAXSIZE];

int currentSize;

cout << "Nhap so luong phan tu cua mang: ";

cin >> currentSize;

int M = INT\_MIN; //Có thể gán M = values[0] rồi đem so sánh với các phần tử còn lại trong mảng

// int M = values[0]

for (int i = 0; i < currentSize; i++) //Vòng lập nhập giá trị cho mảng

{

cin >> values[i];

}

for (int i = 0; i < currentSize; i++) //Vòng lập tìm số lớn nhất mảng

{

if (values[i] > M)

M = values[i];

}

cout << M << " la phan tu lon nhat" << endl;

system("pause");

return 0;

}

//Bài 8.4 Viết chương trình nhập vào một mảng số nguyên(tối đa 100 phần tử), tính và in ra

//tổng và giá trị trung bình của các phần tử trong mảng.

#include<iostream>

using namespace std;

int main8\_4()

{

const int MAXSIZE = 100;

double values[MAXSIZE];

int currentSize;

cout << "Nhap so luong phan tu cua mang: ";

cin >> currentSize;

double Tong = 0;

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

{

cin >> values[i];

}

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

{

Tong += values[i];

}

cout << "Tong la : " << Tong << endl;

cout << "Trung binh tong la : " << (double)Tong / currentSize << endl;

system("pause");

return 0;

}

//Bài 8.5 Viết chương trình nhập vào một mảng số nguyên(tối đa 100 phần tử) và vị trí của

//phần tử cần xóa, xóa phần tử tại vị trí đã nhập và in ra mảng sau khi xóa.

//Cách xóa một vị trí trong mảng

//Vị trí thỏa điều điện tồn tại

//Dùng vòng lập, gán giá trị tại vị trí bằng giá trị ô nằm sau vị trí (phía tay phải)

//Lưu ý vòng lập chạy từ vị trí đến < n phần tử trong mảng trừ đi 1

#include<iostream>

using namespace std;

int main8\_5()

{

const int MAXSIZE = 100;

int values[MAXSIZE];

int currentSize;

cout << "Nhap so luong phan tu cua mang: ";

cin >> currentSize;

int vitri;

cout << "Nhap vi tri can xoa ";

cin >> vitri;

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

{

cin >> values[i];

}

if (vitri >= 0 && vitri < currentSize) //Điều kiện để xóa vị trí ( vị trí phải nằm trong phạm vi có trong mảng)

{

for (int i = vitri; i < currentSize - 1; i++)

{

values[i] = values[i + 1];

}

currentSize--;

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

{

cout << values[i] << " ";

}

}

else

cout << "Nhap sai!!";

system("pause");

return 0;

}

//Bài 8.6 Viết chương trình gồm các hàm sau đây:

// a) Nhập một dãy số nguyên vào mảng arr, số phần tử nhập vào là arrSize :

// void nhap(int arr[], int arrSize);

// b) In dãy số nguyên trong arr, cách nhau bởi dấu phẩy, arrSize là số phần tử của

// dãy :

// void xuat(const int arr[], int arrSize);

// c) Tìm một giá trị x trong mảng arr, trả về true tìm thấy, ngược lại trả về false,

// arrSize là số phần tử của mảng :

// bool tim(const int arr[], int arrSize, int x);

// d) Đảo ngược các phần tử trong mảng arr, arrSize là số phần tử của mảng :

// void daonguoc(int arr[], int arrSize)

// e) Xóa phần tử tại một vị trí p trong mảng arr, arrSize là số phần tử của mảng :

// void xoa(int arr[], int& arrSize, int p)

// f)Thêm phần tử vào cuối mảng

// k)Thêm phần tử vào đầu mảng

//Cách đảo mảng 1 chiều có 2 cách

//Dùng vòng lập xuất mảng ngược

//Đảo trong mảng luôn

#include <iostream>

using namespace std;

#define MAX 100

void nhap(int arr[], int arrSize);

void xuat(const int arr[], int arrSize); //Xuất mảng thì mảng không thay đổi nên dùng const

void xuatDaoNguoc(const int arr[], int arrSize);

bool tim(const int arr[], int arrSize, int x);

void daonguoc(int arr[], int arrSize);

void xoa(int arr[], int& arrSize, int p);

void themphantuCuoi(int arr[], int& arrSize, int m); //Nhớ kiểm tra coi nó có quá giới hạn của mảng khum

// Tính kích thước của mảng với công thức này.

// n = sizeof (mảng\_name) / sizeof (mảng\_name [0])

// Đây là một công thức phổ biến để tìm số lượng phần tử trong

// một mảng, có nghĩa là nó sẽ hoạt động cho các mảng của tất cả dữ liệu

// các loại như int, char, float, v.v.

void themphantuDau(int arr[], int& arrSize, int m);

void themphantuK(int arr[], int& arrSize, int vitri, int q);

int main8\_6() {

int a[MAX], n, x, p, m, k, q, vitri;

cout << "Nhap so phan tu: ";

cin >> n;

nhap(a, n);

cout << "Cac phan tu trong mang la: ";

xuat(a, n);

cout << endl;

cout << "Nhap 1 gia tri muon tim trong mang: ";

cin >> x;

int dem = tim(a, n, x);

if (dem == true)

cout << "Tim thay." << endl;

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

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

daonguoc(a, n);

xuat(a, n);

cout << endl;

//cout << "Mang dao sau khi dung vong lap xuat nguoc : "; //Sau khi mảng đảo thì xuất đảo bị lỗi

//xuatDaoNguoc(a, n);

//cout<<endl;

cout << "Nhap gia tri muon them vao cuoi mang : ";

cin >> m;

themphantuCuoi(a, n, m);

xuat(a, n);

cout << endl;

cout << "Nhap gia tri muon them vao dau mang : ";

cin >> k;

themphantuDau(a, n, k);

xuat(a, n);

cout << endl;

cout << "Nhap vi tri can them vao mang : ";

cin >> vitri;

cout << "Nhap gia tri se them vao : "; cin >> q;

themphantuK(a, n, vitri, q);

xuat(a, n);

cout << endl;

/\*cout << "Nhap vi tri muon xoa: ";

cin >> p;

if (p >= 0 && p < n) {

xoa(a, n, p);

cout << "Mang sau khi xoa la: ";

xuat(a, n);

}

else cout << "Nhap sai.";\*/

system("pause");

return 0;

}

void nhap(int arr[], int arrSize) {

cout << "Nhap cac phan tu cua mang: ";

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

cin >> arr[i];

}

void xuat(const int arr[], int arrSize) {

for (int i = 0; i < arrSize; i++) {

if (i == arrSize - 1) {

cout << arr[i];

break;

}

cout << arr[i] << ", ";

}

}

void xuatDaoNguoc(const int arr[], int arrSize)

{

for (int i = arrSize - 1; i >= 0; i--) {

if (i == arrSize - 1) {

cout << arr[i];

break;

}

cout << arr[i] << ", ";

}

}

bool tim(const int arr[], int arrSize, int x) {

for (int i = 0; i < arrSize; i++) {

if (arr[i] == x)

return true;

}

return false;

}

void daonguoc(int arr[], int arrSize)

{

for (int i = 0; i < arrSize / 2; i++) // Chỉ cần đổi vế đầu thì vế sau tự đổi

//Mượn biến tạm để gán

//T = n - i - 1 (trừ 1 vì trong mảng tính từ 0 đến < n )

{

int tmp = arr[i];

arr[i] = arr[arrSize - i - 1];

arr[arrSize - i - 1] = tmp;

}

}

void xoa(int arr[], int& arrSize, int p) //Hàm xóa 1 phần tử tại vị trí bất kỳ

{

//Cách 1

for (int i = p; i < arrSize - 1; i++)

arr[i] = arr[i + 1];

//Cách 2 cũng như cách 1 mà nó biểu diễn hơi khác 1 tí

//for (int i = p + 1; i < arrSize - 1; i++)

// arr[i - 1] = arr[i];

arrSize--;

}

void themphantuCuoi(int arr[], int& arrSize, int m) //Số lượng phần tử mảng thay đổi nên dùng tham chiếu để lưu số lượng n cho chính xác

{

int size = sizeof(arr) / sizeof(arr[0]);

// Tính kích thước của mảng với công thức này.

// n = sizeof (mảng\_name) / sizeof (mảng\_name [0])

// Đây là một công thức phổ biến để tìm số lượng phần tử trong

// một mảng, có nghĩa là nó sẽ hoạt động cho các mảng của tất cả dữ liệu

// các loại như int, char, float, v.v.

if (size == arrSize) //Số lượng mảng đã đầy (cái đã define)

cout << "Khong the them vao mang\n";

else // Có thể bỏ else và gán trực tiếp cái phía sau

{

arr[arrSize] = m; //Thường thì mảng bắt đầu từ 0 đến arrSize - 1 ; mà do thêm 1 phần tử nên có arrSize trong mảng

arrSize++; //Tăng kích thước mảng lên sau khi thêm phần tử mới vòa mảng

}

}

void themphantuDau(int arr[], int& arrSize, int m)

{

int size;

size = sizeof(arr) / sizeof(arr[0]);

if (size == arrSize)

cout << "Khong the them vao mang\n";

else

{

arrSize++;

for (int i = arrSize - 1; i > 0; i--) //Chừa cái i = 0 để lát gán giá trị

{

arr[i] = arr[i - 1];

}

arr[0] = m;

}

}

void themphantuK(int arr[], int& arrSize, int vitri, int q)

{

int size;

size = sizeof(arr) / sizeof(arr[0]);

if (size == arrSize)

cout << "Khong the them vao mang\n";

else

{

arrSize++;

for (int i = arrSize - 1; i > vitri; i--)

{

arr[i] = arr[i - 1];

}

arr[vitri] = q;

}

}

//XÓA DỊCH VỀ BÊN TRÁI

//THÊM DỊCH VỀ BÊN PHẢI

//Bài 8.11 Viết hàm sắp xếp các phần tử trong mảng tăng dần

#include <iostream>

using namespace std;

void nhap\_mang(int arr[], int n);

void Hoanvi(int& a, int& b);

void Sapxep(int arr[], int n);

void xuat\_mang(int arr[], int n);

int main8\_11()

{

const int AX = 50;

int arr[AX], n;

cout << "Nhap so luong phan tu: ";

cin >> n;

nhap\_mang(arr, n);

Sapxep(arr, n);

cout << "Cac phan tu sau khi duoc sap xep : ";

xuat\_mang(arr, n);

system("pause");

return 0;

}

void nhap\_mang(int arr[], int n) {

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

cout << "Nhap phan tu arr[" << i << "] : ";

cin >> arr[i];

}

}

void Hoanvi(int& a, int& b) {

int tmp = a;

a = b;

b = tmp;

}

void Sapxep(int arr[], int n) {

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

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

if (arr[i] > arr[j]) {

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

}

}

}

}

void xuat\_mang(int arr[], int n) {

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

cout << arr[i] << " ";

}

}

//Bài 8.12 Viết hàm xóa các phần tử trùng nhau trong mảng

#include <iostream>

using namespace std;

void nhapmang(int arr[], int n);

void xuatmang(int arr[], int n);

void hoan\_vi(int& a, int& b);

void sap\_xep(int arr[], int n);

int main8\_12()

{

const int AX = 50;

int arr[AX], n;

cout << "Nhap so luong phan tu: ";

cin >> n;

nhapmang(arr, n);

sap\_xep(arr, n);

cout << "Sau khi xoa cac phan tu trung nhau : ";

xuatmang(arr, n);

system("pause");

return 0;

}

void nhapmang(int arr[], int n) {

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

cout << "Nhap phan tu arr[" << i << "] : ";

cin >> arr[i];

}

}

void hoan\_vi(int& a, int& b) {

int tmp = a;

a = b;

b = tmp;

}

void sap\_xep(int arr[], int n) {

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

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

if (arr[i] > arr[j]) {

hoan\_vi(arr[i], arr[j]);

}

}

}

}

void xuatmang(int arr[], int n) //Cái nào trùng thì không cho xuất

{

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

if (arr[i] != arr[i + 1]) {

cout << arr[i] << " ";

}

}

}

//Bài 1.32 Viết hàm kiểm tra 1 giá trị năm nhận vào có phải là năm nhuần hay không.Viết

//chương trình kiểm tra lại hàm vừa xây dựng.

#include <iostream>

using namespace std;

bool namNhuan(int n);

int main1\_32()

{

int nam;

cout << "Nhap nam tai day : ";

cin >> nam;

if (namNhuan(nam) == true)

cout << nam << " La nam nhuan ! ";

else

cout << nam << " Khong la nam nhuan ! ";

system("pause");

return 0;

}

bool namNhuan(int n)

{

if (n % 400 == 0 || (n % 4 == 0 && n % 100 != 0)) // Công thức tìm năm nhuận

return 1; // return true

return 0; // return false

}

//Bài 1.33 Viết hàm tìm ước chung lớn nhất của 2 số nguyên dương bằng thuật giải Euclid.

//Thuật giải Euclid :

//-Nhận vào : 2 số nguyên dương m và n.

//- Trả về : ước chung lớn nhất của m và n.

//- Thuật giải :

//o Bước 1 : nếu n = 0 thì kết quả là m và kết thúc.Ngược lại sang bước 2.

//o Bước 2 : tính số dư r của phép chia m cho n.

//o Bước 3 : gán n cho m, r cho n.Quay lại bước 1.

//Viết chương trình nhận vào 1 phân số(gồm tử số và mẫu số), xuất kết quả rút gọn

//của phân số(dùng hàm ước chung lớn nhất vừa xây dựng).

#include <iostream>

using namespace std;

int UCLN(int, int);

//int BCNN(int , int );

int main1\_33()

{

int tuso, mauso;

cout << "Tu so : ";

cin >> tuso;

do

{

cout << "Mau so (#0) : ";

cin >> mauso;

} while (mauso == 0);

int usc = UCLN(tuso, mauso);

if (usc != 0)

{

tuso /= usc;

mauso /= usc;

}

cout << "Phan sp sau khi rut gon la : " << tuso << " / " << mauso << endl;

system("pause");

return 0;

}

int UCLN(int m, int n)

{

int r;

while (n != 0)

{

r = m % n;

m = n;

n = r;

}

return m;

}

//int BCNN(int m, int n)

//{

// int result;

// result = m \* n / UCLN(m, n);

// return result;

//}

//Bài 1.34 Viết các hàm sau đây :

//a.Hàm kiểm tra một ký tự có phải là ký tự số hay không ?

//b.Hàm kiểm tra một ký tự có phải là ký tự chữ cái hay không ?

//c.Chương trình nhận vào một chuỗi(string), xuất kết quả chuỗi vừa nhập có bao

//nhiêu ký tự số ? bao nhiêu ký tự chữ ? Lưu ý : sử dụng 2 hàm đã xây dựng ở trên.

#include <iostream>

#include <string>

using namespace std;

bool check1(char);

bool check2(char);

int main1\_34()

{

string s;

cout << "Nhap mot chuoi : ";

getline(cin, s);

int demSo = 0;

int demChu = 0;

for (int i = 0; i < s.length(); i++)

{

if (check2(s[i]))

demSo++;

if (check1(s[i]))

demChu++;

}

/\*char kiemtra;

cout << "Ban nhap du lieu : ";

cin >> kiemtra;

if (check1(kiemtra) == true)

cout << "Du lieu nhap vao la ky tu chu ! \n";

else if (check2(kiemtra) == true)

cout << "Du lieu nhap vao la ky tu so ! \n";

else

cout << "Du lieu nhap vao khong la ky tu chu va ky tu so ! \n";\*/

cout << s << " co " << demSo << " chu so va " << demChu << " chu cai \n";

system("pause");

return 0;

}

bool check1(char kiemtra) // Hàm kiểm tra ký tự chữ

{

if (((kiemtra >= 'A') && (kiemtra <= 'Z')) || ((kiemtra >= 'a') && (kiemtra <= 'z')))

return 1; // true

return 0; // false

}

bool check2(char kiemtra) // Hàm kiểm tra ký tự số

{

if ((kiemtra >= '0') && (kiemtra <= '9'))

return 1; //true

return 0; //false

}

//Bài 1.35 Viết các hàm sau đây :

//a.Hàm kiểm tra một ký tự có phải là ký tự in hoa hay không ?

//b.Hàm kiểm tra một ký tự có phải là ký tự in thường hay không ?

//c.Chương trình nhận vào một chuỗi(string), xuất kết quả chuỗi vừa nhập có bao

//nhiêu ký tự in hoa ? bao nhiêu ký tự in thường ? Lưu ý : sử dụng 2 hàm đã xây

//dựng ở trên

#include <iostream>

#include <string>

using namespace std;

bool check1(char);

bool check2(char);

int main1\_35()

{

string s;

cout << "Nhap mot chuoi : ";

getline(cin, s);

int demInHoa = 0;

int demThuong = 0;

for (int i = 0; i < s.length(); i++)

{

if (check1(s[i]))

demInHoa++;

if (check2(s[i]))

demThuong++;

}

cout << s << " co " << demInHoa << " chu in hoa va " << demThuong << " chu in thuong \n";

system("pause");

return 0;

}

bool check1(char kiemtra) // Hàm kiểm tra ký tự in hoa

{

if (((kiemtra >= 'A') && (kiemtra <= 'Z')))

return 1; // true

return 0; // false

}

bool check2(char kiemtra) // Hàm kiểm tra ký tự in thường

{

if ((kiemtra >= 'a') && (kiemtra <= 'z'))

return 1; //true

return 0; //false

}

//Bài 1.36 Viết hàm nhận vào 3 số ngày, tháng, năm; trả về số ngày cách ngày đầu tiên trong

//năm.Viết chương trình kiểm tra hàm vừa viết.

//Ví dụ :

//3 giá trị ngày - tháng - năm là 1 - 1 - 2006 thì kết quả là 1.

//3 giá trị ngày - tháng - năm là 25 - 12 - 2006 thì kết quả là 359

#include <iostream>

using namespace std;

bool namNhuan(int nam);

int soNgay(int thang, int nam);

int main1\_36()

{

int ngay, thang, nam;

int tongngay = 0;

cout << "Nhap ngay, thang, nam: ";

cin >> ngay >> thang >> nam;

if (nam < 0 || thang < 0 || thang> 12 || ngay < 0 || ngay> 31)

cout << "Khong hop le!!" << endl;

else {

if (thang == 1)

tongngay = ngay;

else

tongngay = 0;

for (int i = 1; i <= thang - 1; i++)

tongngay += soNgay(i, nam);

tongngay += ngay;

cout << "Tong so ngay la: " << tongngay << endl;

}

system("pause");

return 0;

}

bool namNhuan(int nam) {

return(nam % 400 == 0 || (nam % 4 == 0 && nam % 100 != 0));

}

int soNgay(int thang, int nam)

{

int songay = 0;

switch (thang) {

case 1: case 3: case 5: case 7: case 8: case 10: case 12:

songay = 31;

break;

case 4: case 6: case 9: case 11:

songay = 30;

break;

case 2:

if (namNhuan(nam))

songay = 29;

else

songay = 28;

break;

}

return songay;

}

//Đề kt thử Câu 1) Khai báo nguyên mẫu hàm và định nghĩa các hàm sau:

// a) Hàm S(n) nhận vào một số nguyên dương n(n > 0) và trả về trị của đa thức :

// S(n) = 1 + 1 / 2 - 1 / 3 + 1 / 4 – 1 / 5 … + 1 / n(1, 5 điểm)

// b) Hàm SoChuSoLe(n) : nhận vào số nguyên dương n(n > 0) và trả về số các chữ số lẻ có

// trong n. (1, 5 điểm)

// c) Áp dụng hai hàm trên, viết chương trình hiển thị một menu cho phép người dùng chọn

// một trong hai tác vụ(1 hoặc 2), sau đó cho phép nhập số nguyên dương n và hiển thị kết

// quả ra màn hình.Chương trình hiển thị yêu cầu hỏi người dùng có tiếp tục hay không,

// nếu muốn tiếp tục, nhấn ký tự ‘y’ hoặc ‘Y’, màn hình được xóa và hiển thị lại menu cho

// phép người dùng tiếp tục thực hiện thao tác, nếu nhập ký tự bất kỳ khác, chương trình sẽ

// kết thúc(2 điểm)

#include<iostream>

#include<cstdlib>

#include<iomanip>

using namespace std;

double tinhS(int n);

int SoChuSoLe(int n);

int main1\_\_()

{

char tiep;

int chon, n;

do

{

cout << "1. Tinh chu so le cua mot so nguyen\n";

cout << "2. Tinh S(n) = 1 + 1/2 - 1/3 + 1/4 - 1/5 ... + 1/n \n";

cout << "Chon 1 hoac 2 : ";

cin >> chon;

switch (chon)

{

case 1:

cout << "TINH CHU SO LE CUA MOT SO NGUYEN \n";

cout << "Nhap so nguyen : ";

cin >> n;

cout << n << " co " << SoChuSoLe(n) << " so le \n";

break;

case 2:

cout << "Tinh S(n)\n";

cout << "Nhap so nguyen : "; cin >> n;

cout << fixed << setprecision(3) << "S(" << n << ") = " << tinhS(n); //Dùng setprecision thì nhớ khai báo thư viện iomanip

cout << endl;

break;

default:

cout << "Nhap lua chon sai !\n";

}

cout << "Ban muon tiep tuc chon (Y / N) : ";

cin >> tiep;

if (tiep == 'y' || tiep == 'Y' || tiep == 'n' || tiep == 'N')

{

system("cls"); //Lệnh xóa màn hình ; lưu ý nhớ khai báo thư viện cstdlib

continue;

}

else

break;

} while (tiep == 'y' || tiep == 'Y' || tiep == 'n' || tiep == 'N');

system("pause");

return 0;

}

double tinhS(int n)

{

double tong = 1.0;

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

{

if (i % 2 == 0) //Chẵn

{

tong += (1.0 / i);

}

else //Lẻ

{

tong -= (1.0 / i);

}

}

return tong;

}

int SoChuSoLe(int n)

{

int dem = 0;

int sodu;

while (n > 0)

{

sodu = n % 10;

if (sodu % 2 == 1) //Số lẻ

dem++;

n /= 10;

}

return dem;

}

//Đề kt thử Câu 2) Viết chương trình cho phép tạo ngẫu nhiên một mảng số nguyên n phần tử (n được nhập

// từ bàn phím), các số trong mảng chỉ chứa các giá trị từ 1 – 20. (1 điểm)

// a) In các giá trị trong mảng ra màn hình, mỗi phần tử cách nhau dấu phẩy và 1 khoảng trắng (1 điểm)

// b) Xuất kết quả thống kê cho biết mảng có bao nhiêu số chẵn và trung bình cộng các số

// chẵn trong mảng. (1, 5 điểm)

// c) Sắp xếp mảng theo thứ tự : các số lẻ nằm về bên trái có thứ tự giảm dần, các số chẵn nằm

// về bên phải có thứ tự tăng dần(lưu ý : không dùng mảng phụ).In mảng kết quả ra màn

// hình. (1, 5 điểm)

// Ví dụ

// Mảng ban đầu : 6, 5, 8, 12, 17, 8, 4, 3, 1, 5

// Mảng đã sắp xếp : 17, 5, 5, 3, 1, 4, 6, 8, 8, 12

#include<iostream>

#include<ctime> // for time ()

#include <cstdlib> // for srand() and rand()

void cauB(int arr[], int n);

void cauC(int arr[], int n);

void HoanVi(int& x, int& y);

using namespace std;

#define MAX 50

int main1\_\_2()

{

int a[MAX];

int n;

srand(time(0));

cout << "Nhap so phan tu cho mang : ";

cin >> n;

//Thêm phần kiểm tra n

//n <= 0 cout "Nhap sai"

//n > 0 làm tiếp

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

//Nhập mảng ; dùng hàm thì void nhapMang(int arr[], int &n) (hàm nhập nên dùng tham chiếu n)

{

a[i] = 1 + rand() % 20;

}

cout << "Gia tri cua mang là : ";

for (int i = 0; i < n; i++) //Xuất mảng ; dùng hàm thì void xuatMang(int arr[], int n)

{

if (i == n - 1) //cuối mảng

{

cout << a[i] << endl;

}

else

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

}

cauB(a, n);

cauC(a, n);

system("pause");

return 0;

}

void cauB(int arr[], int n)

{

int demchan = 0, tongchan = 0;

double tbChan;

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

{

if (arr[i] % 2 == 0) //Giá trị của mảng là chẵn

{

demchan++;

tongchan += arr[i];

}

}

tbChan = tongchan / demchan;

cout << "Mang co " << demchan << " so chan\n";

cout << "Trung binh cac so chan cua mang la : " << tbChan << endl;

}

void cauC(int arr[], int n)

{

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

{

if (arr[i] % 2 != 0)

{

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

{

if (arr[i] < arr[j] && arr[j] % 2 != 0)

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

}

cout << arr[i] << " ";

}

}

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

{

if (arr[i] % 2 == 0)

{

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

{

if (arr[i] > arr[j] && arr[j] % 2 == 0)

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

}

cout << arr[i] << " ";

}

}

}

void HoanVi(int& x, int& y)

{

int tam;

tam = x;

x = y;

y = tam;

}

//Bài 1.37 Viết hàm cho biết ngày hôm trước của một ngày cho trước

#include <iostream>

using namespace std;

unsigned int tinhSoNgayTrongThang(unsigned int month, unsigned int year);

bool laNamNhuan(unsigned int month);

int main1\_37()

{

unsigned int nDay, nMonth, nYear;

cout << "Nhap ngay, thang, nam tai day : ";

cin >> nDay >> nMonth >> nYear;

// Ngày hôm trước chính là ngày giảm 1 đơn vị

nDay -= 1; // Day = Day - 1

if (nDay == 0)

{

// Ngày sẽ là ngày lớn nhất của tháng trước đó

nMonth -= 1; // Month = Month - 1

if (nMonth == 0)

{

nMonth = 12;

nYear -= 1; // Year = Year - 1

}

nDay = tinhSoNgayTrongThang(nMonth, nYear);

}

cout << "Ngay truoc do la : " << nDay << "-" << nMonth << "-" << nYear << endl;

system("pause ");

return 0;

}

unsigned int tinhSoNgayTrongThang(unsigned int month, unsigned int year)

{

// Tháng có 31 ngày: 1,3,5,7,8,10,12

// Tháng 2: 28, 29 (nhuận) ngày

// Tháng có 30 ngày: 4,6,9,11

unsigned int result;

switch (month)

{

case 1: case 3: case 5: case 7: case 8: case 10: case 12:

result = 31;

break;

case 2:

if (laNamNhuan(year) == true)

{

result = 29;

}

else

{

result = 28;

}

break;

case 4: case 6: case 9: case 11:

result = 30;

break;

}

return result;

}

bool laNamNhuan(unsigned int year)

{

// Kiểm tra năm nhuận

// Là chia hết cho 400 hoặc chia hết cho 4 nhưng không chia hết cho 100

if (year % 400 == 0 || (year % 4 == 0 && year % 100 != 0))

{

return true;

}

return false;

}

//Bài 1.38 Viết hàm cho biết ngày hôm sau của một ngày cho trước

#include <iostream>

using namespace std;

unsigned int tinhSoNgayTrongThang(unsigned int month, unsigned int year);

bool laNamNhuan(unsigned int month);

int main1\_38()

{

unsigned int nDay, nMonth, nYear;

cout << "Nhap ngay, thang, nam tai day : ";

cin >> nDay >> nMonth >> nYear;

// Ngày hôm trước chính là ngày tăng 1 đơn vị

nDay += 1;

if (nDay > tinhSoNgayTrongThang(nMonth, nYear))// Không hợp lệ

{

nDay = 1;

nMonth += 1; // Month = Month + 1

if (nMonth > 12) // Không hợp lệ

{

nMonth = 1;

nYear += 1; // Year = Year + 1

}

}

cout << "Ngay sau do la : " << nDay << "-" << nMonth << "-" << nYear << endl;

system("pause ");

return 0;

}

unsigned int tinhSoNgayTrongThang(unsigned int month, unsigned int year)

{

// Tháng có 31 ngày: 1,3,5,7,8,10,12

// Tháng 2: 28, 29 (nhuận) ngày

// Tháng có 30 ngày: 4,6,9,11

unsigned int result;

switch (month)

{

case 1: case 3: case 5: case 7: case 8: case 10: case 12:

result = 31;

break;

case 2:

if (laNamNhuan(year) == true)

{

result = 29;

}

else

{

result = 28;

}

break;

case 4: case 6: case 9: case 11:

result = 30;

break;

}

return result;

}

bool laNamNhuan(unsigned int year)

{

// Kiểm tra năm nhuận

// Là chia hết cho 400 hoặc chia hết cho 4 nhưng không chia hết cho 100

if (year % 400 == 0 || (year % 4 == 0 && year % 100 != 0))

{

return true;

}

return false;

}

//Bài 1.39 Tính ước chung lớn nhất giữa a và b

#include<iostream>

using namespace std;

int tinhUocChungLonNhat(int a, int b);

int main1\_39()

{

int a, b;

cout << "Nhap hai so a va b : ";

cin >> a >> b;

int UCLN = tinhUocChungLonNhat(a, b);

cout << "Uoc chung lon nhat la : " << UCLN << endl;

system("pause");

return 0;

}

int tinhUocChungLonNhat(int a, int b)

{

int i = b;

while (i != 0)

{

if (a % i == 0 && (b % i == 0))

return i;

i--;

}

}

//Bài 1.40 Chuyển đổi giữa hệ 16 8 10

#include <iostream>

#include <iomanip> // for std::hex, std::oct, std::dec

using namespace std;

int main1\_40()

{

int n;

cout << "Enter n (hexadecimal): ";

cin >> hex >> n; // nhập số thập lục phân (hệ 16)

cout << "Octal: " << oct << n << endl; // xuất số bát phân (hệ 8)

cout << "Decimal: " << dec << n << endl; // xuất số thập phân (hệ 10)

return 0;

}

//Bài 1.41 Chuyển đổi hệ 10 sang hệ 2

#include <iostream>

using namespace std;

int main1\_41()

{

long dec, rem, i = 1, sum = 0;

cout << "Enter the decimal to be converted:";

cin >> dec;

do

{

rem = dec % 2;

sum = sum + (i \* rem);

dec = dec / 2;

i = i \* 10;

} while (dec > 0);

cout << "The binary of the given number is:" << sum << endl;

system("pause");

return 0;

}

// Dùng các hàm thì phải khai báo thư viện

// #include<cmath> (cmath cho c++, math.h cho c) khi dùng sqrt() pow( , ) acos( )

// sqrt() : căn bậc 2 của 1 số ; pow(x,y) : x mũ y (x hệ số, y số mũ)

// #include<iomanip> khi dùng setw ; fixed << setprecision

//abs(x) |x|

//exp(x) e^x , e = 2.718

//pow(x, y) x^y

//sqrt(x) x^1/2 or căn bậc 2 x

//sin(x) sinx, x: radians

//cos(x) cosx, x: radians

//tan(x) tanx, x: radians

//log(x) ln(x),x > 0

//log10(x) log10(x),x > 0

// Hàm acos() trong C / C++ tính cosin trong C và C++

// Cú pháp : acos(data\_type x)

// Trong đó x là giá trị cần truyền vào, có hai dạng đó là Degrees (độ) và Radians

// Chuyển từ độ sang rad : Độ => độ\*pi(3.14)/180 rad

// Chuyển từ rad sang độ : Rad => chuyển pi(3.14) thành 180 rồi tính bình thường độ

//Cú pháp ép kiểu

// static\_cast<kiểu int hay ...>(số cần ép kiểu)

// define tenVietHoa giá trị (không cần dấu ";")

// CÁCH TẠO SỐ NGẪU NHIÊN

#include<iostream>

#include <cstdlib> // for rand() and srand()

#include <ctime> // for time()

using namespace std;

int main3()

{

srand(time(0)); //khai báo thư viện #include<ctime> mới dùng được ; có srand thì số tạo ra mỗi lần đều khác nhau

//Dùng srand(time(0)) hay srand(time(NULL)) đều được

//NULL thì ra số ngẫu nhiên không trùng vs số trước còn 0 thì ngược lạ

cout << "Random = " << rand() << endl;

//rand() là lấy ngẫu nhiên số không bị giớ hạn từ 0 tới RAND\_MAX (32767)

//rand() % 10 số tạo ra trong phạm vi [0;9]

//VD:Muốn random [39;79] thì phân rả ra

// = 39 + rand()%41

//VD:Muốn random [-39;50] thì phân rả ra

//= -39 +rand()%90

system("pause");

return 0;

}

//Toán tử 3 ngôi

// nếu phép toán trước dấu ? đúng thì sẽ trả về kết quả trước dấu : còn nếu sai thì trả về kết quả sau dấu :

//getline(cin, strTen);

//(chép hết dữ liệu từ stdin vào biến strTen)

// cin.getline(strTen,5);

//(cho phép nhập 5-1 kí tự vào xâu strTen)

//KhacNhauGiua\_++x\_x++

#include <iostream>

using namespace std;

int mainLT\_1()

{

// Truong hop dat sau

int x = 10;

cout << x++ << endl; // in ra 10

cout << x << endl; // in ra 11

// Truong hop dat truoc

x = 10;

cout << ++x << endl; // in ra 11

system("pause");

return 1;

}

// ÉP KIỂU

#include <iostream>

#include <iomanip> // for std::setprecision()

using namespace std;

int mainLT\_2()

{

int n{ 75 };

cout << static\_cast<char>(n) << endl;

// in ký tự với mã ASCII 75

char ch{ 'K' };

cout << static\_cast<int>(ch) << endl;

// in mã ASCII của ký tự 'K'

return 0;

}

// Cú pháp ép kiểu static\_cast< kiểu biến >(tên biến);

///////ĐÃ IN /////////////////////////////////////////