// 1. Viết chương trình nhập vào một phân số, rút gọn phân số và xuất kết quả.

#include <bits/stdc++.h>

using *namespace* std;

*class* *PhanSo* {

*private:*

*long* *long* Tuso;

*long* *long* Mauso;

*public:*

        friend *istream*& operator >> (*istream*&, *PhanSo*&);

        friend *ostream*& operator << (*ostream*&, *PhanSo*);

*long* *long* getTuso(){

            return this->Tuso;

        }

*long* *long* getMauso(){

            return this->Mauso;

        }

};

*long* *long* gcd(*long* *long* *a*, *long* *long* *b*){

    if(*b* == 0) return *a*;

    return gcd(*b*, *a*%*b*);

}

*istream*& operator >> (*istream*& *in*, *PhanSo*& *x*){

*in* >> *x*.Tuso >> *x*.Mauso;

    return *in*;

}

*ostream*& operator << (*ostream*& *out*, *PhanSo* *x*){

*long* *long* y1, y2;

*long* *long* UocChung = gcd(*x*.getTuso(), *x*.getMauso());

    y1 = *x*.getTuso()/UocChung;

    y2 = *x*.getMauso()/UocChung;

    if(y2 == 1)

*out* << y1;

    else if(y2 < 0){

        y2 = -y2;

        y1 = -y1;

*out* << y1 << '/' << y2;

    }

    else

*out* << y1 << '/' << y2;

    return *out*;

}

*int* main(){

*PhanSo* x;

    cin >> x;

    if(x.getMauso() == 0)

        cout << "Error! Division by zero.";

    if(x.getTuso() == 0)

        cout << 0;

    else cout << x;

    return 0;

}

// 2. Viết chương trình nhập vào hai phân số, tìm phân số lớn nhất và xuất kết quả.

#include <iostream>

using *namespace* std;

*class* *PhanSo* {

*private:*

*int* TuSo, MauSo;

*public:*

*int* getTuso() {

        return this->TuSo;

    }

*int* getMauso() {

        return this->MauSo;

    }

*void* input();

*void* output();

*PhanSo* operator - (*PhanSo*);

};

*void* compare(*PhanSo*, *PhanSo*);

*int* main() {

*PhanSo* a, b;

    cout << "\nPhan so thu nhat: " << endl;

    a.input();

    cout << "\nPhan so thu hai: " << endl;

    b.input();

    cout << endl;

    compare(a, b);

    cout << endl;

    system("pause");

    return 0;

}

*void* *PhanSo*::input() {

    cout << " Tu so la: ";

    cin >> TuSo;

    cout << " Mau so la: ";

    cin >> MauSo;

    while (MauSo == 0) {

        cout << "Mau so phai khac 0. Mau so moi la: ";

        cin >> MauSo;

    }

}

*void* *PhanSo*::output() {

    if (TuSo % MauSo == 0)

        cout << TuSo / MauSo;

    else

        cout << TuSo << "/" << MauSo << " ";

}

*PhanSo* *PhanSo*::operator - (*PhanSo* *x*) {

*PhanSo* result;

    result.TuSo = TuSo \* *x*.MauSo - *x*.TuSo \* MauSo;

    result.MauSo = *x*.MauSo \* MauSo;

    return result;

}

*void* compare(*PhanSo* *x*, *PhanSo* *y*) {

*PhanSo* subtractResult = *x* - *y*;

*int* temp = subtractResult.getTuso() \* subtractResult.getMauso();

    if (temp > 0) {

        cout << "Phan so lon nhat la ";

*x*.output();

    } else if (temp < 0) {

        cout << "Phan so lon nhat la ";

*y*.output();

    } else

        cout << "Hai phan so bang nhau";

    cout << endl;

}

// 3. Viết chương trình nhập vào hai phân số. Tính tổng, hiệu, tích, thương giữa chúng và xuất kết quả.

#include <iostream>

using *namespace* std;

*class* *PhanSo* {

*private:*

*int* TuSo, MauSo;

*public:*

*int* getTuso() {

        return this->TuSo;

    }

*int* getMauso() {

        return this->MauSo;

    }

*void* rutgon();

    friend *istream*& operator >> (*istream*&, *PhanSo*&);

    friend *ostream*& operator << (*ostream*&, *PhanSo*);

*PhanSo* operator - (*PhanSo*);

*PhanSo* operator + (*PhanSo*);

*PhanSo* operator \* (*PhanSo*);

*PhanSo* operator / (*PhanSo*);

};

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

    if (*b* == 0) return *a*;

    return gcd(*b*, *a* % *b*);

}

*void* *PhanSo*::rutgon() {

*int* ucln = gcd(abs(TuSo), abs(MauSo));

    TuSo /= ucln;

    MauSo /= ucln;

    if (MauSo < 0) {

        TuSo = -TuSo;

        MauSo = -MauSo;

    }

}

*istream*& operator >> (*istream*& *in*, *PhanSo*& *x*){

*in* >> *x*.TuSo >> *x*.MauSo;

    while (*x*.MauSo == 0) {

        cout << "Error!";

        cin >> *x*.MauSo;

    }

    return *in*;

}

*ostream*& operator << (*ostream*& *out*, *PhanSo* *x*){

*x*.rutgon();

    if (*x*.TuSo % *x*.MauSo == 0)

*out* << *x*.TuSo / *x*.MauSo;

    else{

*out* << *x*.TuSo << "/" << *x*.MauSo << " ";

    }

    return *out*;

}

*PhanSo* *PhanSo*::operator - (*PhanSo* *x*) {

*PhanSo* result;

    result.TuSo = TuSo \* *x*.MauSo - *x*.TuSo \* MauSo;

    result.MauSo = *x*.MauSo \* MauSo;

    return result;

}

*PhanSo* *PhanSo*::operator + (*PhanSo* *x*) {

*PhanSo* result;

    result.TuSo = TuSo \* *x*.MauSo + *x*.TuSo \* MauSo;

    result.MauSo = *x*.MauSo \* MauSo;

    return result;

}

*PhanSo* *PhanSo*::operator \* (*PhanSo* *x*) {

*PhanSo* result;

    result.TuSo = TuSo \* *x*.TuSo;

    result.MauSo = *x*.MauSo \* MauSo;

    return result;

}

*PhanSo* *PhanSo*::operator / (*PhanSo* *x*) {

*PhanSo* result;

    result.TuSo = TuSo \* *x*.MauSo;

    result.MauSo = *x*.TuSo \* MauSo;

    if(result.MauSo == 0){

        cout << "Error!";

        exit(1);

    }

    return result;

}

*int* main() {

*PhanSo* a, b;

    cout << "Phan so thu nhat: ";

    cin >> a;

    cout << "Phan so thu hai: ";

    cin >> b;

    cout << "Tong: " << a + b << endl;

    cout << "Hieu: " << a - b << endl;

    cout << "Tich: " << a \* b << endl;

    cout << "Thuong: " << a / b << endl;

    return 0;

}

// 4. Viết chương trình nhập vào một ngày. Tìm ngày kế tiếp và xuất kết quả.

#include <bits/stdc++.h>

using *namespace* std;

*class* *NTN* {

*private:*

*long* *long* day, month, year;

*public:*

    NTN() : day(1), month(1), year(1) {}

    NTN(*long* *long* *d*, *long* *long* *m*, *long* *long* *y*) : day(*d*), month(*m*), year(*y*) {}

    friend *istream*& operator >> (*istream*&, *NTN*&);

    friend *ostream*& operator << (*ostream*&, *NTN*);

*bool* isLeapYear();

*bool* isValidDate();

*NTN* getNextDate();

};

*bool* *NTN*::isValidDate() {

    if (year < 1 || day < 1)

        return false;

    if (month < 1 || month > 12)

        return false;

    if ((month == 1 || month == 3 || month == 5 || month == 7 || month == 8 || month == 10 || month == 12) && (day > 31))

        return false;

    if ((month == 4 || month == 6 || month == 9 || month == 11) && (day > 30))

        return false;

    if ((isLeapYear() && month == 2 && day > 29) || (!isLeapYear() && month == 2 && day > 28))

        return false;

    return true;

}

*bool* *NTN*::isLeapYear(){

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

        return true;

    return false;

}

*NTN* *NTN*::getNextDate() {

*NTN* tempDate(++this->day, this->month, this->year);

    if (tempDate.day > 31) {

        tempDate.day = 1;

        tempDate.month++;

        if (tempDate.month > 12) {

            tempDate.month = 1;

            tempDate.year++;

        }

    }

    else if (tempDate.day > 30) {

        if (tempDate.month == 4 || tempDate.month == 6 || tempDate.month == 9 || tempDate.month == 11)

        {

            tempDate.day = 1;

            tempDate.month++;

        }

    }

    else if (tempDate.month == 2) {

        if (isLeapYear()) {

            if (tempDate.day > 29)

            {

                tempDate.day = 1;

                tempDate.month++;

            }

        }

        else {

            if (tempDate.day > 28)

            {

                tempDate.day = 1;

                tempDate.month++;

            }

        }

    }

    return tempDate;

}

*istream*& operator >> (*istream*& *in*, *NTN*& *x*){

*in* >> *x*.day >> *x*.month >> *x*.year;

    return *in*;

}

*ostream*& operator << (*ostream*& *out*, *NTN* *x*){

*out* << *x*.day << '/' << *x*.month << '/' << *x*.year;

    return *out*;

}

*int* main() {

*NTN* x;

    cin >> x;

    cout << x.getNextDate();

    return 0;

}

// 5. Viết chương trình nhập họ tên, điểm toán, điểm văn của một học sinh. Tính điểm trung bình và xuất kết quả.

#include <bits/stdc++.h>

using *namespace* std;

*class* *HocSinh*{

*private:*

*double* Math, Literature;

*string* Name;

*public:*

    friend *istream*& operator >> (*istream*&, *HocSinh*&);

    friend *ostream*& operator << (*ostream*&, *HocSinh*);

*double* getToan(){

        return this->Math;

    }

*double* getVan(){

        return this->Literature;

    }

};

*double* DTB (*HocSinh* *x*){

*double* Dtb = (*x*.getToan() + *x*.getVan())/2;

    return Dtb;

}

*istream*& operator >> (*istream*& *in*, *HocSinh*& *x*){

    getline(*in*, *x*.Name);

*in* >> *x*.Math >> *x*.Literature;

    return *in*;

}

*ostream*& operator << (*ostream*& *out*, *HocSinh* *x*){

    cout << *x*.Name << " " << DTB(*x*);

    return *out*;

}

*int* main() {

*HocSinh* x;

    cin >> x;

    cout << x;

    return 0;

}

// 4. Xây dựng lớp vận động viên VanDongVien gồm:

* Thuộc tính: hoten (chuỗi ký tự), tuoi (số nguyên), monthidau (chuỗi ký tự), cannang (số thực), chieucao (số thực).
* Phương thức:
  + Thiết lập không tham số.
  + Thiết lập 5 tham số
  + Hủy bỏ
  + Nạp chồng toán tử nhập >>
  + Nạp chồng toán tử xuất <<
  + Nạp chồng toán tử so sánh > (một vận động viên là lớn hơn nếu chiều cao lớn hơn,  
    trong trường hợp chiều cao bằng nhau thì xét cân nặng lớn hơn)

Xây dựng chương trình chính:

* Khai báo p là đối tượng lớp Vandongvien (sử dụng hàm thiết lập 5 tham số), hiển thị thông tin của p ra màn hình.
* Nhập vào một mảng gồm n vận động viên.
* Hiển thị danh sách đã nhập ra màn hình.
* Sắp xếp mảng đã nhập theo thứ tự tăng dần, hiển thị danh sách đã sắp ra màn hình.

#include <bits/stdc++.h>

using *namespace* std;

*class* *VanDongVien* {

*private:*

*string* hoten, monthidau;

*long* *long* tuoi;

*double* cannang, chieucao;

*public:*

        VanDongVien(){

            hoten = monthidau = " ";

            cannang = chieucao = 0;

            tuoi = 0;

        }

        VanDongVien(const *VanDongVien*& *x*){

            this->hoten = *x*.hoten;

            this->monthidau = *x*.monthidau;

            this->cannang = *x*.cannang;

            this->chieucao = *x*.chieucao;

            this->tuoi = *x*.tuoi;

        }

        ~VanDongVien(){}

        friend *istream*& operator >> (*istream* &*in*, *VanDongVien* &*x*){

*in*.ignore();

            getline(*in*, *x*.hoten);

            getline(*in*, *x*.monthidau);

*in* >> *x*.tuoi;

*in* >> *x*.cannang >> *x*.chieucao;

            return *in*;

        }

        friend *ostream*& operator << (*ostream* &*out*, const *VanDongVien* *x*){

*out* << *x*.hoten << endl;

*out* << *x*.monthidau << endl;

*out* << *x*.tuoi << endl;

*out* << *x*.cannang << endl;

*out* << *x*.chieucao << endl;

            return *out*;

        }

*bool* operator > (const *VanDongVien*& *x*) const{

            if(*x*.chieucao == chieucao) return this->cannang > *x*.cannang;

            else return this->chieucao > *x*.chieucao;

        }

*bool* operator < (const *VanDongVien*& *x*) const {

            if (chieucao == *x*.chieucao) return cannang < *x*.cannang;

            return chieucao < *x*.chieucao;

    }

};

*int* main(){

*int* n; cin >> n;

*vector*<*VanDongVien*> v(n);

    for(*int* i = 0; i<n; i++){

        cin >> v[i];

    }

    sort(v.begin(), v.end());

    for(*VanDongVien* x : v){

        cout << x;

    }

    return 0;

}