// 1. Xây dựng lớp biểu diễn khái niệm số phức với hai thành phần dữ liệu thực, ảo và các hàm thành phần xuất, nhập, định giá trị cho số phức, cộng, trừ, nhân, chia hai số phức. Viết chương trình cho phép nhập vào hai số phức, in ra kết quả các phép toán cộng, trừ, nhân, chia hai số phức kể trên. Ví dụ: Cho hai số phức A(a1, a2), B(b1, b2) • A + B = (a1+b1, a2+b2) • A - B = (a1-b1, a2-b2) • A \* B = (a1*b1 – a2*b2, a1*b2+a2*b1) • A / B = ((a1*b1 + a2*b2)/(b1*b1 + b2*b2) , (b1*a2 - a1*b2)/(b1*b1 + b2*b2))

#include <bits/stdc++.h>

using *namespace* std;

*class* *SoPhuc* {

*private:*

*double* Thuc, Ao;

*public:*

        SoPhuc(){

            Thuc = Ao = 0;

        }

        SoPhuc(*double* *x*, *double* *y*){

            this->Thuc = *x*;

            this->Ao = *y*;

        }

*void* input(){

            cout << "Nhap so phuc: ";

            cin >> Thuc >> Ao;

        }

*void* output(){

            cout << "(" << Thuc << ", " << Ao << ")";

        }

*double* getThuc(){

            return this->Thuc;

        }

*double* getAo(){

            return this->Ao;

        }

};

*void* Tong(*SoPhuc* *a*, *SoPhuc* *b*){

*SoPhuc* sum (*a*.getThuc() + *b*.getThuc(), *a*.getAo() + *b*.getAo());

    cout << "A + B = ";

    sum.output();

    cout << endl;

}

*void* Tru(*SoPhuc* *a*, *SoPhuc* *b*){

*SoPhuc* sub (*a*.getThuc() - *b*.getThuc(), *a*.getAo() - *b*.getAo());

    cout << "A - B = ";

    sub.output();

    cout << endl;

}

*void* Nhan(*SoPhuc* *a*, *SoPhuc* *b*){

*SoPhuc* Tich (*a*.getThuc() \* *b*.getThuc() - *a*.getAo() \* *b*.getAo(), *a*.getThuc() \* *b*.getAo() + *b*.getThuc() \* *a*.getAo());

    cout << "A \* B = ";

    Tich.output();

    cout << endl;

}

*void* Chia(*SoPhuc* *a*, *SoPhuc* *b*){

*SoPhuc* Thuong ((*a*.getThuc() \* *b*.getThuc() + *a*.getAo() \* *b*.getAo()) / (*b*.getThuc()\**b*.getThuc() + *b*.getAo()\**b*.getAo()), (*b*.getThuc() \* *a*.getAo() - *a*.getThuc() \* *b*.getAo()) / (*b*.getThuc()\**b*.getThuc() + *b*.getAo()\**b*.getAo()));

    cout << "A / B = ";

    Thuong.output();

    cout << endl;

}

*int* main(){

*SoPhuc* a, b;

    a.input();

    b.input();

    Tong(a,b);

    Tru(a,b);

    Nhan(a,b);

    Chia(a,b);

    return 0;

}

// 2.Làm lại bài số phức với một phương thức thiết lập duy nhất cho phép quan điểm một số thực như một số phức đặc biệt (phần ảo bằng 0). Định nghĩa các phép toán +, -, \*, /, = =, !=, trên số phức. Định nghĩa phép toán << và >> để xuất và nhập dữ liệu vào số phức.

#include <bits/stdc++.h>

using *namespace* std;

*class* *SoPhuc* {

*private:*

*double* Thuc, Ao;

*public:*

        SoPhuc(){

            Thuc = Ao = 0;

        }

        SoPhuc(*double* *x*, *double* *y*){

            this->Thuc = *x*;

            this->Ao = *y*;

        }

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

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

*SoPhuc* operator + (*SoPhuc* *a*){

*SoPhuc* sum (*a*.Thuc + Thuc, *a*.Ao + Ao);

            return sum;

        }

*SoPhuc* operator - (*SoPhuc* *a*){

*SoPhuc* sub (Thuc - *a*.Thuc, Ao - *a*.Ao);

            return sub;

        }

*SoPhuc* operator \* (*SoPhuc* *a*){

*SoPhuc* Tich (*a*.Thuc \* Thuc - *a*.Ao \* Ao, *a*.Thuc \* Ao + Thuc \* *a*.Ao);

            return Tich;

        }

*SoPhuc* operator / (*SoPhuc* *a*){

*double* Mau = *a*.Thuc\**a*.Thuc + *a*.Ao\**a*.Ao;

            if(Mau == 0){

                cout << "Error!" << endl;

                return *SoPhuc*(0,0);

            }

            else {

*SoPhuc* Thuong ((*a*.Thuc \* Thuc + *a*.Ao \* Ao) / (*a*.Thuc\**a*.Thuc + *a*.Ao\**a*.Ao), (*a*.Thuc \* Ao - Thuc \* *a*.Ao) / (*a*.Thuc\**a*.Thuc + *a*.Ao\**a*.Ao));

                return Thuong;

            }

        }

*bool* operator == (*SoPhuc* *a*){

            return (Thuc == *a*.Thuc) && (Ao == *a*.Ao);

        }

*bool* operator != (*SoPhuc* *a*){

            return (Thuc != *a*.Thuc) && (Ao != *a*.Ao);

        }

*double* getThuc(){

            return this->Thuc;

        }

*double* getAo(){

            return this->Ao;

        }

};

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

    cout << "Nhap so phuc: ";

*in* >> *x*.Thuc >> *x*.Ao;

    return *in*;

}

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

    if(*x*.Ao >= 0){

*out* << *x*.Thuc << "+" << *x*.Ao << "i";

    }

    else

*out* << *x*.Thuc << *x*.Ao << "i";

    return *out*;

}

*int* main(){

*SoPhuc* a, b;

    cin >> a >> b;

    cout << a + b << endl;

    cout << a - b << endl;

    cout << a \* b << endl;

    cout << a / b << endl;

    if(a == b) cout << "Hai so phuc bang nhau!";

    if(a != b) cout << "Hai so phuc khac nhau!";

    return 0;

}

/\* 3. Xây dựng lớp Candidate (Thí sinh) gồm các thuộc tính: mã, tên, ngày tháng năm sinh, điểm thi Toán, Văn, Anh và các phương thức cần thiết. Theo đó, hãy xây dựng chương trình:

    - Nhập vào n thí sinh (n do người dùng nhập).

    - In ra thông tin về các thí sinh có tổng điểm lớn hơn 15.\*/

#include <bits/stdc++.h>

using *namespace* std;

*class* *Thisinh* {

*private:*

*string* ma, ten, ns;

*double* toan, van, anh;

*public:*

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

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

*double* gettong() {

        return toan + van + anh;

    }

};

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

    cout << "Nhap ma: ";

*in* >> *x*.ma;

*in*.ignore();

    cout << "Nhap ten: ";

    getline(*in*, *x*.ten);

    cout << "Nhap ngay sinh (dd/mm/yyyy): ";

*in* >> *x*.ns;

    cout << "Nhap diem Toan, Van, Anh: ";

*in* >> *x*.toan >> *x*.van >> *x*.anh;

    return *in*;

}

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

*out* << *x*.ma << " " << *x*.ten << " " << *x*.ns << " "

        << fixed << setprecision(1) << *x*.toan << " " << *x*.van << " " << *x*.anh << " "

        << *x*.gettong() << endl;

    return *out*;

}

*int* main() {

*int* n;

    cout << "Nhap so luong thi sinh: ";

    cin >> n;

*vector*<*Thisinh*> a(n);

    for (*int* i = 0; i < n; i++) {

        cout << "Nhap thong tin thi sinh thu " << i + 1 << ":" << endl;

        cin >> a[i];

    }

    cout << "\nDanh sach thi sinh co tong diem > 15:\n";

    for (*int* i = 0; i < n; i++) {

        if (a[i].gettong() > 15) {

            cout << a[i];

        }

    }

    return 0;

}

// 4. Chuẩn hoá ngày sinh, tên của sinh viên, sắp xếp theo thứ tự GPA giảm dần.

#include <bits/stdc++.h>

using *namespace* std;

*string* chuan(*string* *name*){

*stringstream* ss(*name*);

*string* res = "";

*string* token;

    while(ss >> token){

        res += toupper(token[0]);

        for(*int* i = 1; i < token.length(); i++) res += tolower(token[i]);

        res += " ";

    }

    res.erase(res.length() - 1);

    return res;

}

*class* *SinhVien*{

*public:*

*string* ma, ten, lop, ns;

*double* gpa;

        static *int* cnt;

*public:*

        SinhVien(){

            ma = ten = lop = ns = "";

            gpa = 0;

        }

        friend *bool* operator < (*SinhVien* *a*, *SinhVien* *b*){

            return *a*.gpa < *b*.gpa;

        }

        friend *ostream*& operator << (*ostream*& *out*, const *SinhVien* *a*){

*out* << *a*.ma << " " << *a*.ten << " " << *a*.lop << " " << *a*.ns << " " << fixed << setprecision(2) << *a*.gpa << endl;

            return *out*;

        }

        friend *istream*& operator >> (*istream*& *in*, *SinhVien* &*a*){

            ++cnt;

*string* tmp = to\_string(cnt);

            tmp = *string*(3 - tmp.size(), '0') + tmp;

*a*.ma = "B20DCCN" + tmp;

            scanf("\n");

            getline(*in*, *a*.ten);

*a*.ten = chuan(*a*.ten);

            getline(*in*, *a*.lop);

            getline(*in*, *a*.ns);

*in* >> *a*.gpa;

            if(*a*.ns[2] != '/') *a*.ns = "0" + *a*.ns;

            if(*a*.ns[5] != '/') *a*.ns.insert(3, "0");

            return *in*;

        }

};

*int* *SinhVien*::cnt = 0;

*void* sapxep(*SinhVien* *a*[], *int* *n*){

    sort(*a*, *a* + *n*);

    reverse(*a*, *a* + *n*);

}

*int* main(){

*SinhVien* ds[50];

*int* N, i;

    cin >> N;

    for(i=0;i<N;i++){

        cin >> ds[i];

    }

    sapxep(ds, N);

    for(i=0;i<N;i++){

        cout << ds[i];

    }

    return 0;

}

// 5. In danh sách nhân viên có mã số in tự động bắt đầu từ 0001.

#include <bits/stdc++.h>

using *namespace* std;

*class* *NhanVien*{

*private:*

*string* ten, gt, ns, dc, mst, hd;

*int* ma;

*public:*

        static *int* cnt;

        friend *ostream*& operator << (*ostream* &*out*, *NhanVien* *a*){

*out* << setfill('0') << setw(5) << *a*.ma << " " <<

*a*.ten << " " << *a*.gt << " " << *a*.ns << " " << *a*.dc << " " << *a*.mst << " " << *a*.hd << endl;

            return *out*;

        }

        friend *istream*& operator >> (*istream* &*in*, *NhanVien* &*a*){

            ++cnt;

*a*.ma = cnt;

*in*.ignore();

            getline(*in*, *a*.ten);

*in* >> *a*.gt >> *a*.ns;

*in*.ignore();

            getline(*in*, *a*.dc);

*in* >>*a*.mst >> *a*.hd;

            return *in*;

        }

};

*int* *NhanVien*::cnt = 0;

*int* main(){

*NhanVien* ds[50];

*int* N,i;

    cin >> N;

    for(i=0;i<N;i++) cin >> ds[i];

    for(i=0;i<N;i++) cout << ds[i];

    return 0;

}