**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**ĐẠI HỌC SƯ PHẠM THÀNH PHỐ HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

🙚🙚🙚🕮🙘🙘🙘

**LẬP TRÌNH NÂNG CAO**

**BÀI TẬP QUÁ TRÌNH 01**

Họ và tên : Phạm Nguyệt Anh

MSSV: 46.01.104.005

**Giảng viên: Nguyễn Thị Ngọc Hoa**

Thành phố Hồ Chí Minh, ngày 14 tháng 3 năm 2023

**MỤC LỤC**

[CHƯƠNG 1: QUÁ TẢI TOÁN TỬ 4](#_Toc133272122)

[1. Cơ bản 4](#_Toc133272123)

[1.1. COMPLEX\_NUMBER 4](#_Toc133272124)

[1.2. DATHUC3 6](#_Toc133272125)

[1.3. SODAO 11](#_Toc133272126)

[1.4. SOMOI 13](#_Toc133272127)

[1.5. HCN 15](#_Toc133272128)

[1.6. PHANSO5 17](#_Toc133272129)

[1.7. DATHUC2 18](#_Toc133272130)

[1.8. DATE 21](#_Toc133272131)

[1.9. DTRON 31](#_Toc133272132)

[1.11. SOCHANLE 39](#_Toc133272133)

[1.12. XUATDIEM 42](#_Toc133272134)

[1.13. PSOHONSO 45](#_Toc133272135)

[1.14. TGIAC 49](#_Toc133272136)

[1.15. DATHUC 53](#_Toc133272137)

[1.16. PhanSo2 56](#_Toc133272138)

[1.17. QTaiMang2 59](#_Toc133272139)

[1.18. QtaiMang 61](#_Toc133272140)

[1.19. QtaiPSo 64](#_Toc133272141)

[2. Trung bình 66](#_Toc133272142)

[2.1. COLORED\_POINT 66](#_Toc133272143)

[2.2. Operator01 66](#_Toc133272144)

[2.3. DAOHAM 66](#_Toc133272145)

[CHƯƠNG 2: KHUÔNG HÌNH (TEMPLATE) 70](#_Toc133272146)

[2. Cơ bản 70](#_Toc133272147)

[2.1. TEMPLATESOMOI 70](#_Toc133272150)

[2.2. TEMPLATEHCN 73](#_Toc133272151)

[2.3. TEMP\_M1C 76](#_Toc133272152)

[2.4. TEMPLATE1 77](#_Toc133272153)

[2.5. KhuonHinhTong 81](#_Toc133272154)

[2.7. Template2 87](#_Toc133272155)

[2.8. KhuonHinhSS 90](#_Toc133272156)

[2.9. Template1 91](#_Toc133272157)

[CHƯƠNG 3: CON TRỎ 94](#_Toc133272158)

[3.1. 1718\_2\_LTNC\_Midterm\_01 94](#_Toc133272160)

[3.2. CONTRO1 97](#_Toc133272161)

[3.3. CONTRO2 98](#_Toc133272162)

[3.4. CONTRO3 101](#_Toc133272163)

[3.5. CONTRO5 103](#_Toc133272164)

[3.6. Josephus 105](#_Toc133272165)

[3.7. Pointer\_Array\_01 107](#_Toc133272166)

[3.8. Pointer\_Mang\_2 109](#_Toc133272167)

[3.9. Pointer\_MissingPrime 110](#_Toc133272168)

[CHƯƠNG 4: VECTOR 112](#_Toc133272169)

[4.1. xoa\_vector 112](#_Toc133272172)

[4.2. VECTOR3 113](#_Toc133272173)

[4.3. Vector 114](#_Toc133272174)

[4.4. Vector2 117](#_Toc133272175)

[4.5. Vector1 118](#_Toc133272176)

[CHƯƠNG 5: NHẬP XUẤT FILE 120](#_Toc133272177)

[5.1. ONTAP 120](#_Toc133272183)

[5.2. FSTREAM1 127](#_Toc133272184)

[5.3. fstream3 128](#_Toc133272185)

[5.4. fstream2 129](#_Toc133272186)

[5.5. fstream1 130](#_Toc133272187)

[CHƯƠNG 6: STRING (CHUỖI) 131](#_Toc133272188)

[6.1. Capitalize 131](#_Toc133272195)

[6.2. CHUSOTANCUNG 132](#_Toc133272196)

[6.3. DAONGUOC 133](#_Toc133272197)

[6.4. DAONGUOC1 134](#_Toc133272198)

[6.5. DAONGUOC2 135](#_Toc133272199)

[6.6. DAONGUOC3 136](#_Toc133272200)

[6.7. DEMCHUOI 137](#_Toc133272201)

[6.8. DOIXUNG 139](#_Toc133272202)

[6.9. DATE 140](#_Toc133272203)

[6.10. DAYNP 148](#_Toc133272204)

[6.11. SoPalindrom 151](#_Toc133272205)

[6.12. HAPPYNEWYEAR 152](#_Toc133272206)

[6.13. Hpalindrome 154](#_Toc133272207)

[6.14. NUMCOUNT 155](#_Toc133272208)

[6.15. MINIGAME19.2HOTEN 156](#_Toc133272209)

[6.16. MINIGAME18.2TACHSO 157](#_Toc133272210)

[6.17. LOGIN\_UP 159](#_Toc133272211)

[6.18. LOGIN\_UP2 160](#_Toc133272212)

[6.19. STRINGPBRK 161](#_Toc133272213)

[6.20. XOACHUOI 163](#_Toc133272214)

[6.21. 1718\_2\_LTNC\_Midterm\_02 165](#_Toc133272215)

[CHƯƠNG 7: STACK VÀ QUEUE 166](#_Toc133272216)

[7.1. PALINDROME\_STRING 166](#_Toc133272218)

[7.2. TRUNGHAUTO2 167](#_Toc133272219)

[7.3. TRUNGHAUTO1 169](#_Toc133272220)

[7.4. QUEUE1 171](#_Toc133272221)

[7.5. TT\_HT1 173](#_Toc133272222)

[7.6. DOICS10\_2 174](#_Toc133272223)

[7.7. HAUTO 175](#_Toc133272224)

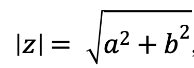
[7.8. STACK1 177](#_Toc133272225)

[7.9. NGOACDUNG 178](#_Toc133272226)

CHƯƠNG 1: QUÁ TẢI TOÁN TỬ

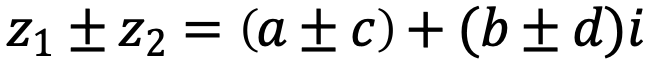
1. Cơ bản
   1. COMPLEX\_NUMBER

Số có dạng z = a + bi, trong đó a, b là số thực, i2= -1, gọi là **số phức**. Trong đó a gọi là phần thực (real), còn b gọi là phần ảo (image).

, gọi là mô-đun của số phức z

Cho hai số phức z1 = a + bi, z2 = c + di

Cộng, trừ số phức:



Xây dựng cấu trúc mô tả một dãy N (0 <= N <= 10,000) số phức với các thao tác sau:

a)     Tính tổng N số phức

b)    Tính hiệu N số phức

c)     Tính mô-đun N số phức

d)    Xuất N số phức theo định dạng: z = a + bi (với a là phần thực và b là phần ảo)

***Yêu cầu****: sử dụng kĩ thuật cấp phát động và con trỏ để quản lý dãy số. Bài làm không đúng yêu cầu sẽ không được tính điểm.*

Input:

-       N dòng, mỗi dòng là 1 cặp a và b mô tả thông tin phần thực và phần ảo của 1 số phức.

Output:

-       Dòng 1: xuất N số phức theo định dạng {a+bi}, mỗi số phức cách nhau 1 khoảng trắng;

-       Dòng 2: giá trị mô-đun của N số phức, lấy tối đa 2 số lẻ ở phần thập phân, mỗi giá trị cách nhau 1 khoảng trắng;

-       Dòng 3: xuất tổng N số phức

Text

Description automatically generated with low confidence

#include <iostream>

#include <math.h>

#include <iomanip>

using namespace std;

struct SoPhuc {

int real;

int image;

void operator = (SoPhuc a) {

real = a.real;

image = a.image;

}

SoPhuc() {

real = image = 0;

}

};

struct Mang {

SoPhuc a[1000];

int n;

SoPhuc& operator[](int i) {

return a[i];

}

};

// So phuc

istream& operator>>(istream& is, SoPhuc& sp) {

is >> sp.real >> sp.image;

return is;

}

ostream& operator<<(ostream& os, SoPhuc sp) {

os << "{" << sp.real << (sp.image > 0 ? "+" : "") << sp.image << "i}" << " ";

return os;

}

//nhap xuat mang

istream& operator>>(istream& is, Mang& m) {

m.n = 0;

while (is >> m.a[m.n++]) {}

m.n--;

return is;

}

ostream& operator<<(ostream& os, Mang m) {

for (int i = 0; i < m.n; i++) {

os << m.a[i];

}

return os;

}

float Module(SoPhuc a) {

float kq;

kq = float(sqrt(pow(a.real, 2) + pow(a.image, 2)));

return kq;

}

SoPhuc operator+(SoPhuc a, SoPhuc b) {

SoPhuc kq;

kq.real = a.real + b.real;

kq.image = a.image + b.image;

return kq;

}

int main()

{

Mang a;

cin >> a;

cout << a << endl;

for (int i = 0; i < a.n; i++) {

cout << fixed << setprecision(2) << Module(a[i]) << " ";

}

cout << endl;

SoPhuc tong;

for (int i = 0; i < a.n; i++) {

tong = tong + a[i];

}

cout << tong;

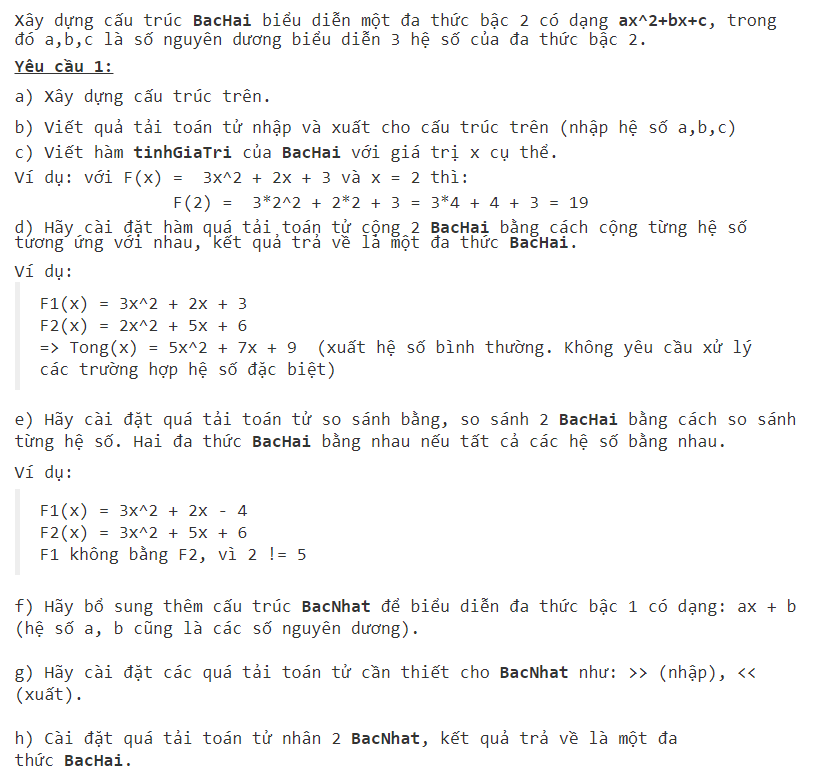
return 0;

}

Text

Description automatically generated

* 1. DATHUC3



Text

Description automatically generated

#include <iostream>

#include <iomanip>

#include <math.h>

using namespace std;

int x =0;

struct bacHai {

int a, b, c;

};

struct bacNhat {

int a, b;

};

istream& operator>>(istream& is, bacHai& bh) {

is >> bh.a >> bh.b >> bh.c;

return is;

}

ostream& operator << (ostream & os, bacHai bh) {

os << bh.a << "x^2";

if (bh.b >= 0) cout << "+"; else "";

os << bh.b << "x";

if (bh.c >= 0) cout << "+"; else "";

os << bh.c << endl;

return os;

}

// bac nhat

istream& operator>>(istream& is, bacNhat& bn) {

is >> bn.a >> bn.b;

return is;

}

ostream& operator << (ostream& os, bacNhat bn) {

os << bn.a << "x";

if (bn.b >= 0) cout << "+"; else "";

os << bn.b;

return os;

}

int tinhGiaTri\_BacHai(bacHai bh) {

return (bh.a \* x \* x + bh.b \* x + bh.c);

}

bacHai operator+(bacHai h1, bacHai h2) {

bacHai kq;

kq.a = h1.a + h2.a;

kq.b = h1.b + h2.b;

kq.c = h1.c + h2.c;

return kq;

}

bacHai operator\*(bacNhat h1, bacNhat h2) {

bacHai kq;

kq.a = h1.a \* h2.a;

kq.b = h1.a \* h2.b + h1.b \* h2.a;

kq.c = h1.b \* h2.b;

return kq;

}

bool operator==(bacHai h1,bacHai h2) {

return (h1.a == h2.a && h1.b == h2.b && h1.c == h2.c);

}

int main()

{

bacHai h1, h2;

bacNhat h3, h4;

cin >> h1 >> h2 >> h3 >> h4;

cin >> x;

cout << h1;

cout << tinhGiaTri\_BacHai(h1) << endl;

cout << h2;

cout << tinhGiaTri\_BacHai(h2) << endl;

cout << h1 + h2;

cout << tinhGiaTri\_BacHai(h1 + h2)<< endl;

bacHai h5 = h3 \* h4;

cout << "(" << h3 << ") \* (" << h4 << ")" << " = " << h5;

cout << tinhGiaTri\_BacHai(h5) << endl;

if (h1 == h2 && h2 == h5) {

cout << "TRUE3";

return 0;

}

if (h1 == h5) {

cout << "TRUE1"; return 0;

}

if (h2 == (h3 \* h4)) {

cout << "TRUE2"; return 0;

}

cout << "FALSE";

return 0;

}

Graphical user interface, application, Word

Description automatically generated

* 1. SODAO

Text

Description automatically generated

#include <iostream>

using namespace std;

struct SoDao {

int a;

void operator=(SoDao sd) {

a = sd.a;

}

};

istream& operator>>(istream& is, SoDao& sd) {

is >> sd.a;

return is;

}

ostream& operator << (ostream& os, SoDao sd) {

os << "[SoDao]" << sd.a << endl;

return os;

}

int DaoSo(SoDao sd) {

int tmp;

int dao = 0 ;

while (sd.a > 0) {

tmp = sd.a % 10;

dao = (dao \* 10) + tmp;

sd.a = sd.a / 10;

}

return dao;

}

bool operator>(SoDao sd1, SoDao sd2) {

return (DaoSo(sd1) > DaoSo(sd2));

}

SoDao operator+(SoDao sd1, SoDao sd2) {

SoDao kq;

kq = sd1 + sd2;

return kq;

}

int tong(SoDao sd1, SoDao sd2) {

return DaoSo(sd1) + DaoSo(sd2);

}

int main()

{

SoDao a, b;

cin >> a >> b;

cout << a << b;

if (a > b) {

cout << "YES" << endl;

}

else

cout << "NO" << endl;

cout << DaoSo(a) + DaoSo(b);

return 0;

}

Graphical user interface

Description automatically generated with low confidence

* 1. SOMOI

Text

Description automatically generated

#include <iostream>

#include <math.h>

using namespace std;

struct SoMoi {

int A;

void operator = (SoMoi sm) {

A = sm.A;

}

};

istream& operator >> (istream& is, SoMoi& sm) {

is >> sm.A;

return is;

}

ostream& operator << (ostream& os, SoMoi sm) {

os << "[SoMoi] " << sm.A << endl;

return os;

}

int tongChuSo(SoMoi sm) {

int dem = 1;

while (sm.A >= 10) {

sm.A /= 10;

dem++;

}

return dem;

}

int tinhTong(SoMoi sm1) {

int tong = 0;

while (sm1.A > 0) {

tong += sm1.A % 10;

sm1.A /= 10;

}

return tong;

}

bool operator> (SoMoi a, SoMoi b) {

return tinhTong(a) > tinhTong(b);

}

SoMoi operator+ (SoMoi sm1, SoMoi sm2) {

SoMoi sm3;

sm3.A = tinhTong(sm1) + tinhTong(sm2);

return sm3;

}

int main()

{

SoMoi a, b;

cin >> a >> b;

cout << a << b;

if (a > b) {

cout << "true" << endl;

}

else

cout << "false" << endl;

cout << a + b;

return 0;

}

Graphical user interface, application

Description automatically generated

* 1. HCN

Graphical user interface, text

Description automatically generated

#include <iostream>

using namespace std;

struct HCN {

int a, b;

};

istream & operator >> (istream &is, HCN &hcn);

ostream & operator << (ostream &os, HCN hcn);

float ChuVi(HCN hcn);

bool operator < (HCN hcn1, HCN hcn2);

float operator + (HCN hcn, float x);

ostream & operator << (ostream &os, HCN hcn){

os << "[HCN]" << hcn.a << "," << hcn.b;

return os;

}

istream & operator >> (istream &is, HCN &hcn){

is >> hcn.a >> hcn.b;

return is;

}

float ChuVi(HCN hcn){

return (hcn.a + hcn.b) \* 2;

}

bool operator < (HCN hcn1, HCN hcn2){

return ChuVi(hcn1) < ChuVi(hcn2);

}

float operator + (HCN hcn, float x){

return ChuVi(hcn) + x;

}

int main(){

HCN h1, h2;

cin >> h1;

cin >> h2;

cout << h1 << endl;

cout << h2 << endl;

if (h1 <h2){

cout << "true";

}

else cout << "false";

return 0;

}

Graphical user interface, text, application, chat or text message

Description automatically generated

* 1. PHANSO5

Graphical user interface, text, application, email

Description automatically generated

#include <iostream>

using namespace std;

struct PhanSo {

int tu, mau;

};

istream& operator >> (istream& is, PhanSo& a) {

is >> a.tu >> a.mau;

return is;

}

ostream& operator << (ostream& os, PhanSo a) {

os << a.tu << "/" << a.mau;

return os;

}

PhanSo operator++ (PhanSo a) {

a.tu += 1;

return a;

}

PhanSo operator-- (PhanSo a) {

a.tu -= 1;

return a;

}

int main()

{

PhanSo a;

string b;

cin >> a;

cin >> b;

cout << a << endl;

if (b == "++") {

cout << ++a;

}else

if (b == "--") {

cout << --a;

}

}

Graphical user interface, application

Description automatically generated with medium confidence

* 1. DATHUC2

Graphical user interface, text, application, email

Description automatically generated

Text

Description automatically generated

#include <iostream>

#include <math.h>

using namespace std;

int x;

struct BacNhat {

int a, b;

int& operator= (BacNhat x) {

x.a = a;

x.b = b;

}

};

int tinhCong(BacNhat h1) {

return (h1.a \* x + h1.b);

}

istream& operator >> (istream& is, BacNhat& bn) {

is >> bn.a >> bn.b;

return is;

}

ostream& operator << (ostream& os, BacNhat bn) {

os << bn.a << "x+" << bn.b;

return os;

}

BacNhat operator+ (BacNhat h1, BacNhat h2) {

BacNhat kq;

kq.a = h1.a + h2.a;

kq.b = h1.b + h2.b;

return kq;

}

bool operator == (BacNhat h1, BacNhat h2) {

if (h1.a + h1.b == h2.a + h2.b)

return true;

return false;

}

int main()

{

BacNhat h1, h2;

cin >> h1 >> h2 >> x;

cout << h1 << "\n" << h2 << endl;

cout << h1 << "+" << h2 << "=" << h1 + h2 << endl;

cout << tinhCong(h1)<< endl;

cout << tinhCong(h2) << endl;

if (h1 == h2)

cout << "TRUE";

else

cout << "FALSE";

return 0;

}

Graphical user interface

Description automatically generated with medium confidence

* 1. DATE

Text, letter

Description automatically generated

Text

Description automatically generated

#include <iostream>

#include <string>

using namespace std;

struct Date

{

int month, day, year;

};

istream& operator>>(istream& inp, Date& d)

{

inp >> d.day >> d.month >> d.year;

return inp;

}

ostream& operator<<(ostream& out, Date d)

{

if (d.day < 10) out << 0 << d.day;

else out << d.day;

out << "/";

if (d.month < 10) out << 0 << d.month;

else out << d.month;

out << "/";

out << d.year;

return out;

}

bool LeapYear(Date d)

{

return (d.year % 4 == 0 && d.year % 100 != 0) || (d.year % 400 == 0);

}

int ndayofYear(Date d)

{

int x = 0, i;

for (i = 1; i < d.month; i++)

{

switch (i)

{

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

{

x += 31;

break;

}

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

{

x += 30;

break;

}

case 2:

{

if (LeapYear(d))

{

x += 29;

}

else

{

x += 28;

}

break;

}

}

}

return x + d.day;

}

string ndayofWeek(Date d)

{

string s;

int n = d.year;

int c = ndayofYear(d);

int t1 = (n - 1) / 4;

int t2 = (n - 1) / 100;

int t3 = (n - 1) / 400;

int k = (n - 1) + t1 - t2 + t3 + c;

switch (k % 7)

{

case 0:

{

s = "Sunday";

break;

}

case 1:

{

s = "Monday";

break;

}

case 2:

{

s = "Tuesday";

break;

}

case 3:

{

s = "Wednesday";

break;

}

case 4:

{

s = "Thursday";

break;

}

case 5:

{

s = "Friday";

break;

}

case 6:

{

s = "Saturday";

break;

}

}

return s;

}

int ndayofMonth(Date d)

{

int nday;

switch (d.month)

{

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

{

nday = 31;

break;

}

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

{

nday = 30;

break;

}

case 2:

{

if (LeapYear(d))

{

nday = 29;

}

else

{

nday = 28;

}

break;

}

}

return nday;

}

Date nextDay(Date d)

{

Date nextd;

if (d.day == ndayofMonth(d))

{

if (d.month == 12)

{

nextd.day = 1;

nextd.month = 1;

nextd.year = d.year + 1;

}

else

{

nextd.day = 1;

nextd.month = d.month + 1;

nextd.year = d.year;

}

}

else

{

nextd.day = d.day + 1;

nextd.month = d.month;

nextd.year = d.year;

}

return nextd;

}

bool operator==(Date d1, Date d2)

{

return d1.day == d2.day && d1.month == d2.month && d1.year == d2.year;

}

void likendayofWeek(Date d1, Date d2)

{

ndayofWeek(d1) == ndayofWeek(d2) ? cout << "TRUE" : cout << "FALSE";

}

bool operator<(Date d1, Date d2)

{

if (d1.year < d2.year)

{

return true;

}

else if (d1.year == d2.year)

{

if (d1.month < d2.month)

{

return true;

}

else if (d1.month == d2.month)

{

if (d1.day < d2.day) return true;

else return false;

}

else return false;

}

else return false;

}

int operator-(Date d1, Date d2)

{

int y;

if (d1.year == d2.year)

{

if (d1 < d2)

{

return ndayofYear(d2) - ndayofYear(d1);

}

else if (d1 == d2) return 0;

else

{

return ndayofYear(d1) - ndayofYear(d2);

}

}

if (d1.year < d2.year)

{

if (LeapYear(d1))

{

y = 366 - ndayofYear(d1);

}

else y = 365 - ndayofYear(d1);

for (int i = d1.year + 1; i < d2.year; i++)

{

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

{

y += 366;

}

else y += 365;

}

return y + ndayofYear(d2);

}

if (d1.year > d2.year)

{

if (LeapYear(d2))

{

y = 366 - ndayofYear(d2);

}

else y = 365 - ndayofYear(d2);

for (int i = d2.year + 1; i < d1.year; i++)

{

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

{

y += 366;

}

else y += 365;

}

return y + ndayofYear(d1);

}

}

int main()

{

Date x1, x2;

cin >> x1;

cin >> x2;

cout << x1 << " " << ndayofWeek(x1) << " " << ndayofYear(x1) << " " << nextDay(x1) << " ";

LeapYear(x1) == true ? cout << "TRUE" : cout << "FALSE";

cout << endl;

cout << x2 << " " << ndayofWeek(x2) << " " << ndayofYear(x2) << " " << nextDay(x2) << " ";

LeapYear(x2) == true ? cout << "TRUE" : cout << "FALSE";

cout << endl;

likendayofWeek(x1, x2);

cout << endl;

if (x1 < x2) cout << "1 < 2";

else if (x1 == x2) cout << "1 = 2";

else cout << "1 > 2";

cout << endl;

cout << x1 - x2;

}

Graphical user interface, text, application, Word

Description automatically generated

* 1. DTRON

Text, letter

Description automatically generated

A picture containing graphical user interface

Description automatically generated

#include <bits/stdc++.h>

using namespace std;

struct Diem {

int td, hd;

};

struct DTRON {

Diem o;

double r;

};

// diem

istream& operator>>(istream& is, Diem& d) {

is >> d.td >> d.hd;

return is;

}

ostream& operator << (ostream& os, Diem d) {

os << "(" << d.td << "," << d.hd << ")";

return os;

}

// duong tron

istream& operator>>(istream& is, DTRON& dt) {

is >> dt.o >> dt.r;

return is;

}

ostream& operator << (ostream& os, DTRON dt) {

os << dt.o << " " << dt.r << endl;

return os;

}

double tinhCV(DTRON dt) {

return (2 \* 3.14 \* dt.r);

}

bool operator==(DTRON dt1, DTRON dt2) {

return tinhCV(dt1) == tinhCV(dt2);

}

bool operator<(DTRON dt1, DTRON dt2) {

return tinhCV(dt1) < tinhCV(dt2);

}

double tinhDT(DTRON dt) {

return 3.14 \* dt.r \* dt.r;

}

double operator+(DTRON dt1, DTRON dt2) {

return tinhDT(dt1) + tinhDT(dt2);

}

//quá tải toán tử trừ ( - ) để tính khoảng cách giữa 2 điểm

Diem operator-(Diem a, Diem b) {

Diem d;

d.td = a.td - b.td;

d.hd = a.hd - b.hd;

return d;

}

double tinhKC(Diem a, Diem b) {

double d;

d = sqrt(pow((b.td - a.td), 2) + pow((b.hd - a.hd), 2));

return d;

}

//Viết quá tải so sánh bằng ( == ) để kiểm tra 2 điểm trùng nhau

bool operator==(Diem a, Diem b) {

return (a.td == b.td && a.hd == b.hd);

}

void VTTD(DTRON dt1, DTRON dt2) {

double d2;

d2 = tinhKC(dt1.o, dt2.o);

if (d2 > abs(dt1.r - dt2.r) && d2 < dt1.r + dt2.r) {

cout << "C" << endl; // cat nhau

}

if (d2 == abs(dt1.r - dt2.r)) {

cout << "TXT" << endl; // tiep xuc trong

}

if (d2 == (dt1.r + dt2.r)) {

cout << "TXN" << endl;

}

if (d2 > (dt1.r + dt2.r)) {

cout << "NN" << endl;

}

if (d2 < abs(dt1.r - dt2.r) && d2 != 0) {

cout << "DN" << endl;

}

if (d2 == 0) {

cout << "DT" << endl; // dong tam

}

}

int main()

{

DTRON dt1, dt2;

cin >> dt1 >> dt2;

cout << dt1 << dt2;

double dt3 = tinhDT(dt1) + tinhDT(dt2);

cout << roundf(dt3 \* 1000) / 1000 << endl;

if (tinhCV(dt1) == tinhCV(dt2)) {

cout << "1 = 2" << endl;

}

if (tinhCV(dt1) < tinhCV(dt2)) {

cout << "1 < 2" << endl;

}

if (tinhCV(dt2) < tinhCV(dt1)) {

cout << "1 > 2" << endl;

}

VTTD(dt1, dt2);

return 0;

}

Graphical user interface, application, Word

Description automatically generated

* 1. DTHANG

Text, application

Description automatically generated

Text

Description automatically generated

A picture containing background pattern

Description automatically generated

#include <bits/stdc++.h>

using namespace std;

struct DTHANG {

double a, b, c;

DTHANG() {

a = b = c = 0;

}

};

// duong thang

istream& operator>>(istream& is, DTHANG& dt) {

is >> dt.a >> dt.b >> dt.c;

return is;

}

ostream& operator << (ostream& os, DTHANG dt) {

// a

if (dt.a == 0) {

os << "";

}

else if (dt.a == 1) {

os << "x";

}

else os << dt.a << "x";

// b

if (dt.b == 0) {

os << "";

}else if (dt.b == 1) {

os << (dt.a == 0 ? "": "+") << "y";

}

else if (dt.b > 1) {

os << (dt.a == 0 ? "" : "+") << dt.b << "y";

}

else {

os << (dt.a == 0 ? "" : "") << dt.b << "y";

}

// c

if (dt.c == 0) {

os << "=0" << endl;

}

else if (dt.c > 0) {

os << "+" << dt.c << "=0" << endl;

}

else {

os << "" << dt.c << "=0" << endl;

}

return os;

}

bool operator == (DTHANG d1, DTHANG d2) {

return d1.a == d2.a && d1.b == d2.b && d1.c == d2.c;

}

char VTTD(DTHANG d1, DTHANG d2) {

int D = d1.a \* d2.b - d2.a \* d1.b;

int Dx = d2.c \* d1.b - d1.c \* d2.b;

int Dy = d2.a \* d1.c - d1.a \* d2.c;

if (D == 0) {

if (Dx == Dy && Dx == 0) {

return 'T';

}

else if (Dx != 0 || Dy != 0) {

return 'S';

}

}

if (d1.a \* d2.a + d1.b \* d2.b == 0) {

return 'V';

}

return 'C';

}

int UCLN(int a, int b) {

if (a < 0)

a = -a;

if (b < 0)

b = -b;

if (a % b == 0)

return b;

return UCLN(b, a % b);

}

int main()

{

DTHANG d1, d2;

cin >> d1 >> d2;

cout << d1 << d2;

cout << VTTD(d1, d2) << endl;

int D = d1.a \* d2.b - d2.a \* d1.b;

int Dx = d2.c \* d1.b - d1.c \* d2.b;

int Dy = d2.a \* d1.c - d1.a \* d2.c;

if (VTTD(d1, d2) == 'C'|| VTTD(d1,d2) == 'V') {

if (Dx % D == 0) {

cout << "(" << Dx / D << ",";

}

else if ((Dx < 0 && D < 0) || (Dx > 0 && D < 0)) {

cout << "(" << -Dx / UCLN(Dx, D) << "/" << -D / UCLN(Dx, D) << ",";

}

else {

cout << "(" << Dx / UCLN(Dx, D) << "/" << D / UCLN(Dx, D) << ",";

}

if (Dy % D == 0) {

cout << Dy / D << ")";

}

else if ((Dy < 0 && D < 0) || (Dy > 0 && D < 0)) {

cout << -Dy / UCLN(Dy, D) << "/" << -D / UCLN(Dy, D) << ")";

}

else {

cout << Dy / UCLN(Dy, D) << "/" << D / UCLN(Dy, D) << ")";

}

}

else if (VTTD(d1, d2) == 'S') {

double Sx, Sy;

if (d1.a != 0) {

Sx = double(-d1.c / d1.a);

Sy = 0;

}

else {

Sx = 0;

Sy = double(-d1.c / d1.b);

}

double KC = abs(d2.a \* Sx + d2.b \* Sy + d2.c) / sqrt((pow(d2.a, 2)) + pow(d2.b, 2));

cout << roundf(KC \* 1000) / 1000;

}

else {

cout << "0";

}

return 0;

}

Graphical user interface, application, Word

Description automatically generated

* 1. SOCHANLE

Text, letter

Description automatically generated

A picture containing background pattern

Description automatically generated

#include <bits/stdc++.h>

using namespace std;

struct SoChanLe {

string a;

};

istream& operator >> (istream& is, SoChanLe& a) {

is >> a.a;

return is;

}

ostream& operator <<(ostream& os, SoChanLe& a) {

for (int i = 0; i < a.a.size(); i += 2) {

os << a.a[i];

}

return os;

}

bool operator< (SoChanLe a, SoChanLe b) {

int A = 0;

int B = 0;

for (int i = 0; i < a.a.size(); i += 2) {

A = A \* 10 + (a.a[i] - 48);

}

for (int i = 0; i < b.a.size(); i += 2) {

B = B \* 10 + (b.a[i] - 48);

}

return A < B;

}

int tongThanhPhan(SoChanLe a, int type = 0) {

int tong = 0;

if (type == 0) {

for (int i = 0; i < a.a.size(); i+= 2) {

tong += (a.a[i] - 48);

}

}

else {

for (int i = 1; i < a.a.size(); i+= 2) {

tong += (a.a[i] - 48);

}

}

return tong;

}

int main() {

SoChanLe a, b;

int type;

cin >> a >> b >> type;

cout << a << "\n" << b << endl;

if (a < b) {

cout << "true" << endl;

}else

cout << "false" << endl;

cout << tongThanhPhan(a, type) << endl;

cout << tongThanhPhan(b, type) << endl;

return 0;

}

Graphical user interface, application, Word

Description automatically generated

* 1. XUATDIEM

Graphical user interface, text, application

Description automatically generated

#include <iostream>

using namespace std;

struct Diem {

int x, y;

void operator = (Diem a) {

x = a.x;

y = a.y;

}

Diem() {

x = 0;

y = 0;

}

};

istream& operator >> (istream& is, Diem& d) {

is >> d.x >> d.y;

return is;

}

ostream& operator << (ostream& os, Diem d) {

os << "(" << d.x << "," << d.y << ")" << endl;

return os;

}

bool operator==(Diem d1, Diem d2) {

if (d1.x == d2.x && d1.y == d2.y)

return true;

return false;

}

bool operator< (Diem d1, Diem d2) {

if (d1.x < d2.x) {

return true;

}

else if (d1.x == d2.x) {

if (d1.y < d2.y)

return true;

return false;

}

if (d1.x > d2.x)

return false;

}

Diem operator+ (Diem a, Diem b) {

Diem kq;

kq.x = a.x + b.x;

kq.y = a.y + b.y;

return kq;

}

int main()

{

Diem a[100];

int n = 0;

while (cin >> a[n]) {

n++;

}

Diem max = a[0];

Diem sum;

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

sum = sum + a[i];

if (max < a[i])

max = a[i];

}

cout << max;

cout << sum;

return 0;

}

Graphical user interface, application, Word

Description automatically generated

* 1. PSOHONSO

Text

Description automatically generated

Table

Description automatically generated

#include <iostream>

using namespace std;

struct PhanSo {

int tu, mau;

};

struct HonSo {

int hs;

PhanSo ps;

void operator= (HonSo a) {

hs = a.hs;

ps = a.ps;

}

};

istream& operator >> (istream& is, PhanSo& a) {

is >> a.tu >> a.mau;

return is;

}

ostream& operator << (ostream& os, PhanSo a) {

os << a.tu << "/" << a.mau;

return os;

}

bool operator!= (PhanSo a, PhanSo b) {

if (a.tu == b.tu && a.mau == b.mau)

return false;

return true;

}

//HON SO

istream& operator >> (istream& is, HonSo& a) {

is >> a.hs >> a.ps;

return is;

}

ostream& operator << (ostream& os, HonSo a) {

os << a.hs << " " << a.ps;

return os;

}

int ucln(int tu, int mau) {

while (mau > 0) {

int x = tu % mau;

tu = mau;

mau = x;

}

return tu;

}

PhanSo rutGonPS(PhanSo a) {

int x = ucln(a.tu, a.mau);

a.tu /= x;

a.mau /= x;

return a;

}

PhanSo operator+ (PhanSo a, PhanSo b) {

PhanSo kq;

kq.tu = a.tu \* b.mau + a.mau \* b.tu;

kq.mau = a.mau \* a.mau;

return rutGonPS(kq);

}

PhanSo operator- (PhanSo a, PhanSo b) {

PhanSo kq;

kq.tu = a.tu \* b.mau - a.mau \* b.tu;

kq.mau = a.mau \* a.mau;

return rutGonPS(kq);

}

HonSo operator-(HonSo a) { return a; }

HonSo operator+ (HonSo a) {

return a;

}

PhanSo chuyenPS(HonSo a) {

PhanSo kq;

kq.tu = a.hs \* a.ps.mau + a.ps.tu;

kq.mau = a.ps.mau;

return rutGonPS(kq);

}

HonSo rutGonHS(HonSo a) {

chuyenPS(a);

int x = a.ps.tu / a.ps.mau;

a.hs += x;

int y = a.ps.tu % a.ps.mau;

a.ps.tu = y;

a.ps = rutGonPS(a.ps);

return a;

}

bool operator!= (HonSo a, HonSo b) {

return !(a.hs == b.hs && a.ps.tu == b.ps.tu && a.ps.mau == b.ps.mau);

}

int tongCacThanhPhan(HonSo a) {

int tong;

tong = a.hs + a.ps.tu + a.ps.mau;

return tong;

}

bool operator > (HonSo a, HonSo b) {

return tongCacThanhPhan(a) > tongCacThanhPhan(b);

}

int main()

{

HonSo a, b;

string c;

cin >> a >> b >> c;

if (a != b) {

cout << "TRUE" << endl;

}

else

cout << "FALSE" << endl;

//

if (a > b) {

cout << "TRUE"<< endl;

}

else

cout << "FALSE" << endl;

//

if (c == "true") {

cout << rutGonHS(a) << "\n" << rutGonHS(b);

}

else if (c == "false")

cout << chuyenPS(a) << "\n" << chuyenPS(b);

return 0;

}

Graphical user interface, application, Word

Description automatically generated

* 1. TGIAC

Text, letter

Description automatically generated

#include <iostream>

#include <math.h>

#include <cmath>

using namespace std;

struct Diem {

int x, y;

void operator = (Diem a) {

x = a.x;

y = a.y;

}

Diem() {

x = 0;

y = 0;

}

};

istream& operator >> (istream& is, Diem& d) {

is >> d.x >> d.y;

return is;

}

ostream& operator << (ostream& os, Diem d) {

os << "(" << d.x << "," << d.y << ")" << " ";

return os;

}

bool operator==(Diem a, Diem b) {

if (a.x == b.x && a.y == b.y)

return true;

return false;

}

double tinhKhoangCach(Diem a, Diem b) {

int d;

d = sqrt ((b.x - a.x) \* (b.x - a.x) + (b.y - a.y) \* (b.y - a.y));

return d;

}

struct TamGiac {

Diem a,b,c;

void operator= (TamGiac tg) {

a = tg.a;

b = tg.b;

c = tg.c;

}

};

istream& operator >> (istream& is, TamGiac& d) {

is >> d.a >> d.b >> d.c;

return is;

}

ostream& operator << (ostream& os, TamGiac d) {

os << d.a << d.b << d.c;

return os;

}

double tinhChuViTG(TamGiac tg) {

double P;

double AB = tinhKhoangCach(tg.a, tg.b);

double AC = tinhKhoangCach(tg.a, tg.c);

double BC = tinhKhoangCach(tg.b, tg.c);

P = AB + AC + BC;

return P;

}

/\*TamGiac operator+ (TamGiac tg1, TamGiac tg2) {

double P3;

P3 = tinhChuViTG(tg1) + tinhChuViTG(tg2);

return tg1;

}\*/

bool operator< (TamGiac a, TamGiac b) {

return tinhChuViTG(a) < tinhChuViTG(b);

}

bool kiemtra(Diem a, Diem b) {

if (a.y == b.y && a.x == b.x)

return true;

return false;

}

bool operator==(TamGiac tg1, TamGiac tg2) {

if (kiemtra(tg1.a, tg2.a) || kiemtra(tg1.a, tg2.b) || kiemtra(tg1.a, tg2.c) &&

kiemtra(tg1.b, tg2.a) || kiemtra(tg1.b, tg2.b) || kiemtra(tg1.b, tg2.c) &&

kiemtra(tg1.c, tg2.a) || kiemtra(tg1.c, tg2.b) || kiemtra(tg1.c, tg2.c))

return true;

return false;

}

int main()

{

TamGiac tg1, tg2;

cin >> tg1 >> tg2;

cout << tg1 << "\n" << tg2 << endl;

if (tg1 < tg2) {

cout << "TRUE" << endl;

}

else {

cout << "FALSE" << endl;

}

if (tg1 == tg2) {

cout << "TRUE" << endl;

}

else

cout << "FALSE";

return 0;

}

Graphical user interface, application, Word

Description automatically generated

* 1. DATHUC

Text

Description automatically generated

Graphical user interface, text, application

Description automatically generated with medium confidence

#include <iostream>

#include <math.h>

using namespace std;

int x;

struct BacNhat {

int a, b;

int& operator= (BacNhat x) {

x.a = a;

x.b = b;

}

};

int tinhCong(BacNhat h1) {

return (h1.a \* x + h1.b);

}

istream& operator >> (istream& is, BacNhat& bn) {

is >> bn.a >> bn.b;

return is;

}

ostream& operator << (ostream& os, BacNhat bn) {

os << bn.a << "x+" << bn.b;

return os;

}

BacNhat operator+ (BacNhat h1, BacNhat h2) {

BacNhat kq;

kq.a = h1.a + h2.a;

kq.b = h1.b + h2.b;

return kq;

}

bool operator == (BacNhat h1, BacNhat h2) {

if (h1.a + h1.b == h2.a + h2.b)

return true;

return false;

}

int main()

{

BacNhat h1, h2;

cin >> h1 >> h2 >> x;

cout << h1 << "\n" << h2 << endl;

cout << h1 << "+" << h2 << "=" << h1 + h2 << endl;

cout << tinhCong(h1)<< endl;

cout << tinhCong(h2) << endl;

if (h1 == h2)

cout << "TRUE";

else

cout << "FALSE";

return 0;

}

Graphical user interface, application, Word

Description automatically generated

* 1. PhanSo2

Text

Description automatically generated

#include <iostream>

#include <cmath>

using namespace std;

struct PhanSo {

int tu, mau;

void operator= (PhanSo a) {

tu = a.tu ;

mau = a.mau;

}

PhanSo() {

tu = 0;

mau = 1;

}

};

int ucln(int tu, int mau) {

while (mau > 0) {

int x = tu % mau;

tu = mau;

mau = x;

}

return tu;

}

struct Mang {

PhanSo a[1000];

int n;

PhanSo& operator[] (int i) {

return a[i];

}

};

PhanSo rutGon(PhanSo a){

int x = ucln(a.tu, a.mau);

PhanSo rg;

rg.tu = a.tu /= x;

rg.mau = a.mau /= x;

return rg;

}

istream& operator >> (istream& is, PhanSo& a) {

is >> a.tu >> a.mau;

return is;

}

ostream& operator << (ostream& os, PhanSo a) {

os << a.tu << "/" << a.mau;

return os;

}

istream& operator>> (istream& is, Mang& m) {

m.n = 0;

while (is >> m.a[m.n++]) {}

m.n--;

return is;

}

ostream& operator<< (ostream& os, Mang m) {

for (int i = 0 ; i < m.n ; i++) {

os << m[i] << endl;

}

return os;

}

bool operator== (PhanSo a, PhanSo b){

return a.tu\*b.mau == a.mau\*b.tu;

}

bool operator!= (PhanSo a, PhanSo b){

return !(a == b);

}

PhanSo operator+ (PhanSo a, PhanSo b) {

PhanSo kq;

kq.tu = a.tu \* b.mau + a.mau \* b.tu;

kq.mau = a.mau \* b.mau;

return rutGon(kq);

}

int main()

{

Mang m;

cin >> m;

PhanSo tong;

for (int i = 0 ; i < m.n ; i++) {

tong = tong + m[i];

}

cout << tong;

return 0;

}

Graphical user interface

Description automatically generated with low confidence

* 1. QTaiMang2

Graphical user interface, text, application, email

Description automatically generated

#include <iostream>

using namespace std;

struct M1C {

int t;

int a[50];

int& operator [] (int i) {

return a[i];

}

void operator = (M1C m) {

t = m.t;

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

a[i] = m[i];

}

};

istream& operator >> (istream& is, M1C &m);

ostream& operator << (ostream& os, M1C m);

M1C operator +(M1C m1, M1C m2);

int main() {

M1C m1, m2;

cin >> m1 >> m2;

M1C m3;

m3=m1+m2;

cout << m3;

return 0;

}

istream& operator >> (istream& is, M1C &m) {

is >> m.t;

for (int i = 0; i < m.t; i++) {

is >> m[i];

}

return is;

}

ostream& operator << (ostream& os, M1C m) {

for (int i = 0; i < m.t; i++) {

os << m[i] << " ";

}

return os;

}

M1C operator + (M1C m1, M1C m2){

M1C m3;

if (m1.t < m2.t){

m3=m2;

for (int i=0; i < m1.t;i++){

m3[i]=m3[i]+m1[i];

}

}

else {

m3=m1;

for (int i =0;i<m2.t;i++){

m3[i]=m3[i]+m2[i];

}

}

return m3;

}

Graphical user interface, application

Description automatically generated

* 1. QtaiMang

Graphical user interface, text, application, email

Description automatically generated#include <iostream>

#include <math.h>

using namespace std;

struct Mang {

int sl;

int a[100];

int& operator = (Mang m) {

m.sl = sl;

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

m.a[i] = a[i];

}

}

int operator[] (int i) {

return a[i];

}

};

istream& operator >> (istream& is, Mang& m) {

is >> m.sl;

for (int i = 0; i < m.sl; i++) {

is >> m.a[i];

}

return is;

}

ostream& operator << (ostream& os, Mang m) {

os << m.sl << endl;

for (int i = 0; i < m.sl; i++) {

os << m.a[i] << " ";

}

return os;

}

Mang operator+ (Mang m1, Mang m2) {

Mang kq;

if (m1.sl <= m2.sl) {

kq.sl = m2.sl;

for (int i = 0; i < kq.sl; i++) {

kq.a[i] = m1.a[i] + m2.a[i];

}

cout << kq;

}

else {

kq.sl = m1.sl;

}

return kq;

}

bool operator==(Mang m1, Mang m2) {

if (m1.sl != m2.sl)

return false;

else {

for (int i = 0; i < m1.sl; i++) {

if (m1.a[i] == m2.a[i])

return true;

}

return false;

}

}

bool operator !=(Mang m1, Mang m2) {

return !(m1 == m2);

}

int main() {

Mang m1,m2;

cin >> m1 >> m2;

if (m1 == m2) {

cout << "yes";

}

else

cout << "no" << endl;

return 0;

}

Diagram

Description automatically generated with medium confidence

* 1. QtaiPSo

Graphical user interface, text, application, email

Description automatically generated

#include <iostream>

#include<cmath>

#include<math.h>

using namespace std;

struct PhanSo {

int tu;

int mau;

};

int ucln(int tu, int mau) {

while (mau > 0) {

int x = tu % mau;

tu = mau;

mau = x;

}

return tu;

}

istream& operator >> (istream& is, PhanSo& a) {

is >> a.tu >> a.mau;

return is;

}

ostream& operator << (ostream& os, PhanSo a) {

os << a.tu << "/" << a.mau;

return os;

}

PhanSo rutGon(PhanSo ps) {

int x = ucln(ps.tu,ps.mau) ;

ps.tu /= x;

ps.mau /= x;

return ps;

}

PhanSo operator+ (PhanSo a, PhanSo b) {

PhanSo kq;

kq.tu = a.tu \* b.mau + a.mau \* b.tu;

kq.mau = a.mau \* b.mau;

return rutGon(kq);

}

bool operator == (PhanSo a, PhanSo b) {

return (a.tu \* b.mau == a.mau \* b.mau);

}

bool operator != (PhanSo a, PhanSo b) {

return !(a == b);

}

int main()

{

PhanSo a, b;

cin >> a >> b;

cout << a + b << endl;

return 0;

}

Text

Description automatically generated with medium confidence

1. Trung bình
   1. COLORED\_POINT
   2. Operator01
   3. DAOHAM

Text

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

#include <iostream>

#include <fstream>

using namespace std;

struct DaThuc

{

int n;

int a[100];

DaThuc()

{

n = 0;

a[0]=0;

}

int& operator [](int n)

{

return a[n];

}

};

istream& operator >> (istream& is, DaThuc& p)

{

is >> p.n;

for (int i = p.n; i >= 0; i--)

{

is >> p[i];

}

return is;

}

ostream& operator << (ostream& os, DaThuc p)

{

if (p[p.n] != 0)

{

if (p.n > 1)

{

if (p[p.n] == 1)

os << "x^" << p.n;

else if (p[p.n] == -1)

os << "-x^" << p.n;

else

os << p[p.n] << "x^" << p.n;

}

else if (p.n == 1)

{

if (p[p.n] == 1)

os << "x";

else if (p[p.n] == -1)

os << "-x";

else

os << p[p.n] << "x";

}

else if (p.n == 0)

{

os << p[0];

return os;

}

}

else if(p.n == 0)

os << p[0];

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

{

if (p[i] > 0)

{

if (p[i] != 1)

os << "+" << p[i] << "x^" << i;

else

os << "+" << "x^" << i;

}

else if (p[i] < 0)

{

if (p[i] != -1)

os << p[i] << "x^" << i;

else

os << "-x^" << i;

}

else

continue;

}

if (p[1] != 0 && p.n >1)

{

if (p[1] > 0)

{

if (p[1] != 1)

os << "+" << p[1] << "x";

else

os << "+" << "x";

}

else if (p[1] < 0)

{

if (p[1] != -1)

os << p[1] << "x";

else

os << "-" << "x";

}

}

if (p[0] != 0)

{

if (p[0] > 0)

os << "+" << p[0];

else if (p[0] < 0)

os << p[0];

}

return os;

}

DaThuc DaoHam(DaThuc p)

{

DaThuc result;

if (p.n > 0)

{

result.n = p.n - 1;

for (int i = result.n; i >= 0; i--)

{

result[i] = p[i + 1] \* (i + 1);

}

}

return result;

}

int main()

{

DaThuc test;

cin >> test;

cout << test << endl;

DaThuc DH1 = DaoHam(test);

DaThuc DH2 = DaoHam(DH1);

cout << DH1 << endl;

cout << DH2;

return 0;

}

Text

Description automatically generated

CHƯƠNG 2: KHUÔNG HÌNH (TEMPLATE)

1. Cơ bản
3. 1. TEMPLATESOMOI

Table

Description automatically generated

#include <iostream>

using namespace std;

struct SoMoi {

long gt;

};

long tongSoMoi(SoMoi sm);

istream& operator >> (istream& is, SoMoi& sm);

ostream& operator << (ostream& os, SoMoi sm);

bool operator > (SoMoi sm1, SoMoi sm2);

long operator + (SoMoi sm, long n);

template<typename T>

struct Mang {

T q[1000];

int n;

};

template <typename T>

istream& operator >> (istream& is, Mang<T>& m) {

T x; m.n = 0;

while (is >> x) {

m.q[m.n++] = x;

}

return is;

}

template <typename T>

T timMax(Mang<T> m) {

T max = m.q[0];

for (int i = 1; i < m.n; i++) {

if (m.q[i] > max) {

max = m.q[i];

}

}

return max;

}

int main() {

char ch;

cin >> ch;

if (ch == 'N') {

Mang<int> N;

cin >> N;

int max = timMax(N), dem = 0, tong = 0;

cout << max << endl;

for (int i = 0; i < N.n; i++) {

if (N.q[i] == max) {

dem++;

}

tong += N.q[i];

}

cout << dem << endl;

cout << tong;

}

else if (ch == 'M') {

Mang<SoMoi> M;

cin >> M;

SoMoi max = timMax(M);

int dem = 0, tong = 0;

cout << max << endl;

for (int i = 0; i < M.n; i++) {

if (tongSoMoi(M.q[i]) == tongSoMoi(max)) {

dem++;

}

tong = M.q[i] + tong;

}

cout << dem << endl;

cout << "[SoMoi] " << tong;;

}

return 0;

}

long tongSoMoi(SoMoi sm) {

long tong = 0; long x;

while (sm.gt > 0) {

x = sm.gt % 10;

sm.gt /= 10;

tong += x;

}

return tong;

}

istream& operator >> (istream& is, SoMoi& sm) {

is >> sm.gt;

return is;

}

ostream& operator << (ostream& os, SoMoi sm) {

os << "[SoMoi] " << sm.gt;

return os;

}

bool operator > (SoMoi sm1, SoMoi sm2) {

return (tongSoMoi(sm1) > tongSoMoi(sm2));

}

long operator + (SoMoi sm, long n) {

return (tongSoMoi(sm) + n);

}

Graphical user interface, application

Description automatically generated

* 1. TEMPLATEHCN

Table

Description automatically generated

#include <iostream>

#include <cmath>

#include <algorithm>

#include <string>

#include <stdio.h>

using namespace std;

struct hcn {

float dai, rong;

void operator =(hcn h) {

dai = h.dai;

rong = h.rong;

}

};

istream& operator >> (istream& is, hcn& h) {

is >> h.dai >> h.rong;

return is;

}

ostream& operator << (ostream& os, hcn h) {

os << "[HCN] " << h.dai << "," << h.rong;

return os;

}

float chuvi(hcn h) {

return (h.dai + h.rong) \* 2;

}

bool operator < (hcn h1, hcn h2) {

return (chuvi(h1) < chuvi(h2));

}

template <class T>

struct Mang {

T a[100];

int n;

T& operator [] (int i) {

return a[i];

}

};

template <class T>

T TimMin(Mang <T> h) {

T Min = h[0];

for (int i = 0; i < h.n; i++) {

if (h[i] < Min) Min = h[i];

}

return Min;

}

float operator + (hcn h1, float a) {

return (chuvi(h1) + a);

}

int main() {

Mang <int> k;

Mang <hcn> h;

string s;

cin >> s;

if (s == "N") {

k.n = 0;

while (cin >> k[k.n]) {

k.n++;

}

cout << TimMin(k) << endl;

float s1 = 0;

for (int i = 0; i < k.n; i++) {

s1 = s1 + k[i];

}

cout << s1;

}

if (s == "H") {

h.n = 0;

while (cin >> h[h.n]) {

h.n++;

}

cout << TimMin(h) << endl;

float s2 = 0;

for (int i = 0; i < h.n; i++) {

s2 = s2 + chuvi(h[i]);

}

printf("%.1f", s2);

}

return 0;

}

A picture containing waterfall chart

Description automatically generated

* 1. TEMP\_M1C

Text

Description automatically generated

#include <iostream>

using namespace std;

template <typename Temp>

struct M1C {

int n;

Temp a[100];

void operator= (M1C A) {

n = A.n;

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

a[i] = A[i];

}

}

Temp& operator[] (int i) {

return a[i];

}

M1C() {

n = 0;

}

};

M1C<int> a;

template <typename Temp>

void xuat(M1C<Temp> a) {

for (int i = 0; i < a.n; i++) {

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

}

}

int tong(M1C<int> a) {

int Tong = a[0];

for (int i = 1; i < a.n; i++) {

Tong = Tong + a[i];

}

return Tong;

}

template <typename Temp>

void xuat() {

cout << tong(a);

}

template <typename Temp>

void nhap(M1C<Temp>& a) {

a.n = 0;

int b;

while (cin >> b) {

a[a.n++] = b;

}

xuat<Temp>();

}

int main() {

nhap<int>(a);

return 0;

}

Graphical user interface, application

Description automatically generated

* 1. TEMPLATE1

Text

Description automatically generated with medium confidence

Background pattern

Description automatically generated with low confidence

#include <bits/stdc++.h>

using namespace std;

class Oxy {

protected:

int x, y;

public:

Oxy(int a = 0, int b = 0) {

x = a;

y = b;

}

Oxy(const Oxy& a) {

x = a.x;

y = a.y;

}

~Oxy() {}

double operator- (Oxy a) {

return sqrt(pow(a.x - x, 2) + pow(a.y - y, 2));

}

bool operator< (Oxy a) {

return x < a.x || (x == a.x && y < a.y);

}

friend istream& operator>> (istream& is, Oxy& a) {

is >> a.x >> a.y;

return is;

}

friend ostream& operator<< (ostream& os, Oxy a) {

os << "(" << a.x << "," << a.y << ")";

return os;

}

};

class Oxyz {

protected:

int x, y, z;

public:

Oxyz(int a = 0, int b = 0, int c = 0) {

x = a;

y = b;

z = c;

}

Oxyz(const Oxyz& a) {

x = a.x;

y = a.y;

z = a.z;

}

~Oxyz() {}

double operator- (Oxyz a) {

return sqrt(pow(a.x - x, 2) + pow(a.y - y, 2) + pow(a.z - z, 2));

}

bool operator< (Oxyz a) {

return x < a.x || (x == a.x && y < a.y) || (x == a.x && y == a.y && z < a.z);

}

friend istream& operator>> (istream& is, Oxyz& a) {

is >> a.x >> a.y >> a.z;

return is;

}

friend ostream& operator<< (ostream& os, Oxyz a) {

os << "(" << a.x << "," << a.y << "," << a.z << ")";

return os;

}

};

template <typename Temp>

bool up(Temp a, Temp b) {

return a < b;

}

template <typename Temp>

bool down(Temp a, Temp b) {

return b < a;

}

template <typename Temp>

class Array {

protected:

int n;

Temp a[1001];

public:

int getn() { return n; }

void setn(int x) { n = x; }

Array() {

n = 0;

}

Temp& operator[] (int i) {

return a[i];

}

void Up() {

sort(a, a + n, up<Temp>);

}

void Down() {

sort(a, a + n, down<Temp>);

}

friend istream& operator>> (istream& is, Array<Temp>& ar) {

is >> ar.n;

for (int i = 0; i < ar.n; i++) {

is >> ar[i];

}

return is;

}

friend ostream& operator<< (ostream& os, Array<Temp> ar) {

for (int i = 0; i < ar.n; i++) {

os << ar[i] << " ";

}

return os;

}

double maxOfTwo() {

double temp = 0;

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

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

if (temp < a[i] - a[j])

temp = a[i] - a[j];

}

}

return roundf(temp \* 1000) / 1000;

}

};

int main() {

string x;

Array<Oxy> a;

Array<Oxyz> b;

while (cin >> x) {

if (x == "Oxy") {

cin >> a[a.getn()];

a.setn(a.getn() + 1);

}

else {

cin >> b[b.getn()];

b.setn(b.getn() + 1);

}

}

a.Up();

b.Down();

cout << a << endl << b << endl;

cout << a.maxOfTwo() << endl;

cout << b.maxOfTwo();

return 0;

}

Word

Description automatically generated

* 1. KhuonHinhTong

Table

Description automatically generated

#include <bits/stdc++.h>

using namespace std;

template <typename Temp>

struct Mang {

int n;

Temp A[100];

Temp& operator[] (int i) {

return A[i];

}

Mang() {

n = 0;

}

};

struct PhanSo {

int Tu, Mau;

void operator= (PhanSo p) {

Tu = p.Tu;

Mau = p.Mau;

}

};

int UCLN(int a, int b) {

while (b > 0) {

int r = a % b;

a = b;

b = r;

}

return a;

}

PhanSo rutGon(PhanSo S) {

int a = UCLN(S.Tu, S.Mau);

S.Tu /= a;

S.Mau /= a;

return S;

}

istream& operator>> (istream& is, PhanSo& p) {

is >> p.Tu >> p.Mau;

return is;

}

ostream& operator<< (ostream& os, PhanSo p) {

os << p.Tu << "/" << p.Mau;

return os;

}

PhanSo operator+ (PhanSo A, PhanSo B) {

PhanSo S;

S.Tu = A.Tu \* B.Mau + A.Mau \* B.Tu;

S.Mau = A.Mau \* B.Mau;

return rutGon(S);

}

template <typename Temp>

Temp tong(Mang<Temp> a) {

Temp S = a[0];

for (int i = 1; i < a.n; i++) {

S = S + a[i];

}

return S;

}

int main() {

Mang<int> mangA;

Mang<PhanSo> mangB;

char Type;

while (cin >> Type) {

switch (Type) {

case 'a': {

int X;

cin >> X;

mangA[mangA.n++] = X;

break;

}

case 'b': {

PhanSo ps;

cin >> ps;

mangB[mangB.n++] = ps;

break;

}

}

}

if (mangA.n == 0)

cout << "khong co";

else

cout << tong<int>(mangA);

cout << endl;

if (mangB.n == 0)

cout << "khong co";

else

cout << tong<PhanSo>(mangB);

cout << endl;

return 0;

}

A picture containing waterfall chart

Description automatically generated

* 1. Graphical user interface, text

     Description automatically generatedKhuonHinh3

A picture containing calendar

Description automatically generated

#include <bits/stdc++.h>

using namespace std;

struct PhanSo {

int Tu, Mau;

void operator= (PhanSo p) {

Tu = p.Tu;

Mau = p.Mau;

}

PhanSo() {

Tu = 0;

Mau = 1;

}

};

istream& operator>> (istream& is, PhanSo& p) {

is >> p.Tu >> p.Mau;

return is;

}

ostream& operator<< (ostream& os, PhanSo p) {

os << p.Tu << "/" << p.Mau;

return os;

}

bool operator> (PhanSo A, PhanSo B) {

return A.Tu \* B.Mau > A.Mau \* B.Tu;

}

template <typename Temp>

struct Mang {

int n;

Temp a[100];

void operator= (Mang A) {

n = A.n;

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

a[i] = A[i];

}

}

Temp& operator[] (int i) {

return a[i];

}

Mang() {

n = 0;

}

};

template <typename Temp>

istream& operator>> (istream& is, Mang<Temp>& m) {

m.n = 0;

for (int i = 0; i < m.n; i++) {

is >> m[i];

}

return is;

}

template <typename Temp>

ostream& operator<< (ostream& os, Mang<Temp> m) {

for (int i = 0; i < m.n; i++) {

os << m[i];

}

return os;

}

template <typename Temp>

Temp min(Mang<Temp> m) {

Temp Min = m[0];

for (int i = 0; i < m.n; i++) {

if (Min > m[i]) {

Min = m[i];

}

}

return Min;

}

template <typename Temp>

void xuat(Mang<Temp> m) {

if (m.n == 0)

cout << "khong co\n";

else

cout << min<Temp>(m) << endl;

}

int main() {

Mang<int> m1;

Mang<float> m2;

Mang<PhanSo> m3;

char Type;

while (cin >> Type) {

if (Type == 'a')

cin >> m1[m1.n++];

else {

if (Type == 'b')

cin >> m2[m2.n++];

else

cin >> m3[m3.n++];

}

}

xuat<int>(m1);

xuat<float>(m2);

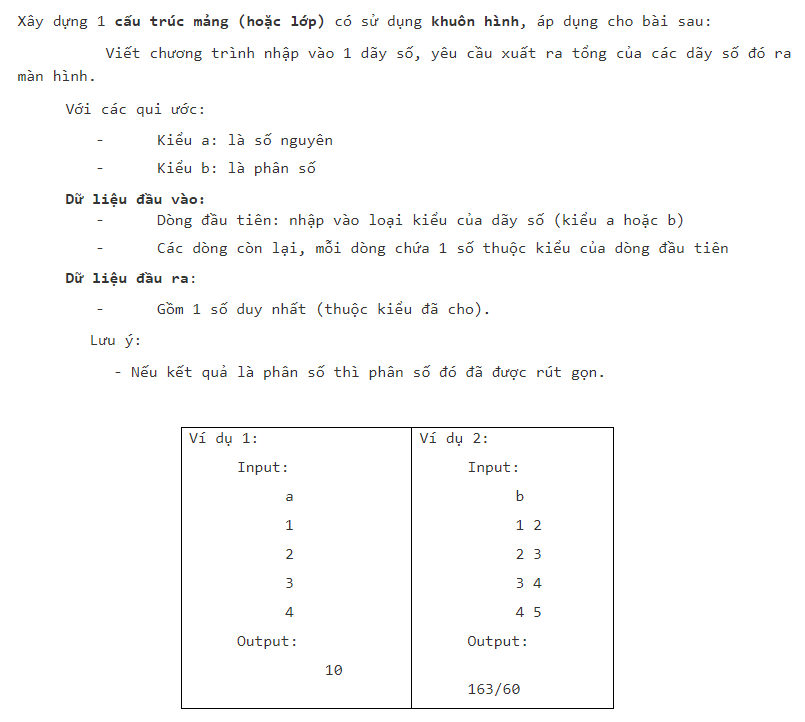
xuat<PhanSo>(m3);

return 0;

A picture containing graphical user interface

Description automatically generated}

* 1. Template2



#include <bits/stdc++.h>

using namespace std;

template <typename Temp>

struct Mang {

int n;

Temp a[100];

Temp& operator[] (int i) {

return a[i];

}

Mang() {

n = 0;

}

};

struct PhanSo {

int Tu, Mau;

void operator= (PhanSo p) {

Tu = p.Tu;

Mau = p.Mau;

}

};

istream& operator>> (istream& is, PhanSo& p) {

is >> p.Tu >> p.Mau;

return is;

}

ostream& operator<< (ostream& os, PhanSo p) {

os << p.Tu << "/" << p.Mau;

return os;

}

int UCLN(int a, int b) {

while (b > 0) {

int r = a%b;

a = b;

b = r;

}

return a;

}

PhanSo rutGon(PhanSo A) {

int a = UCLN(A.Tu, A.Mau);

A.Tu /= a;

A.Mau /= a;

return A;

}

PhanSo operator+ (PhanSo A, PhanSo B) {

PhanSo Tong;

Tong.Tu = A.Tu\*B.Mau + A.Mau\*B.Tu;

Tong.Mau = A.Mau \* B.Mau;

return rutGon(Tong);

}

template <typename Temp>

Temp tongPhanTu(Mang<Temp> mang) {

Temp Tong = mang[0];

for (int i = 1 ; i < mang.n ; i++) {

Tong = Tong + mang[i];

}

return Tong;

}

int main() {

char Type;

cin >> Type;

switch (Type) {

case 'a': {

Mang<int> mangA;

int x;

while (cin >> x) {

mangA[mangA.n++] = x;

}

cout << tongPhanTu<int>(mangA);

break;

}

case 'b': {

Mang<PhanSo> mangB;

PhanSo ps;

while (cin >> ps) {

mangB[mangB.n++] = ps;

}

cout << tongPhanTu<PhanSo>(mangB);

break;

}

}

return 0;

}

A picture containing graphical user interface

Description automatically generated

* 1. KhuonHinhSS

Text

Description automatically generated

#include <iostream>

#include <math.h>

#include <iomanip>

using namespace std;

struct PhanSo {

int Tu, Mau;

void operator= (PhanSo p) {

Tu = p.Tu;

Mau = p.Mau;

}

PhanSo() {

Tu = 0;

Mau = 1;

}

};

istream& operator>> (istream& is, PhanSo& p) {

is >> p.Tu >> p.Mau;

return is;

}

ostream& operator<< (ostream& os, PhanSo p) {

os << p.Tu << "/" << p.Mau;

return os;

}

bool operator== (PhanSo A, PhanSo B) {

return A.Tu \* B.Mau == A.Mau \* B.Tu;

}

template <typename Temp>

void nhap() {

Temp a, b;

cin >> a >> b;

if (a == b)

cout << "true";

else

cout << "false";

}

int main() {

char Type;

cin >> Type;

switch (Type) {

case 'a':

nhap<int>();

break;

case 'b':

nhap<float>();

break;

case 'c':

nhap<PhanSo>();

break;

}

return 0;

}

Graphical user interface, application

Description automatically generated

* 1. Template1

Text

Description automatically generated with medium confidence

#include <bits/stdc++.h>

using namespace std;

struct PhanSo {

int Tu, Mau;

void operator= (PhanSo p) {

Tu = p.Tu;

Mau = p.Mau;

}

};

istream& operator>> (istream& is, PhanSo& p) {

is >> p.Tu >> p.Mau;

return is;

}

ostream& operator<< (ostream& os, PhanSo p) {

os << p.Tu << "/" << p.Mau;

}

bool operator< (PhanSo A, PhanSo B) {

return double(A.Tu/A.Mau) < double(B.Tu/B.Mau);

}

template <typename Temp>

Temp Max(Temp a, Temp b, Temp c) {

Temp max = a;

if (max < b)

max = b;

if (max < c)

max = c;

return max;

}

template <typename Temp>

void nhap() {

Temp a, b, c;

cin >> a >> b >> c;

cout << Max(a, b, c);

}

int main() {

char Char;

cin >> Char;

switch (Char) {

case 'a':

nhap<int>();

break;

case 'b':

nhap<float>();

break;

case 'c':

nhap<PhanSo>();

break;

}

return 0;

}

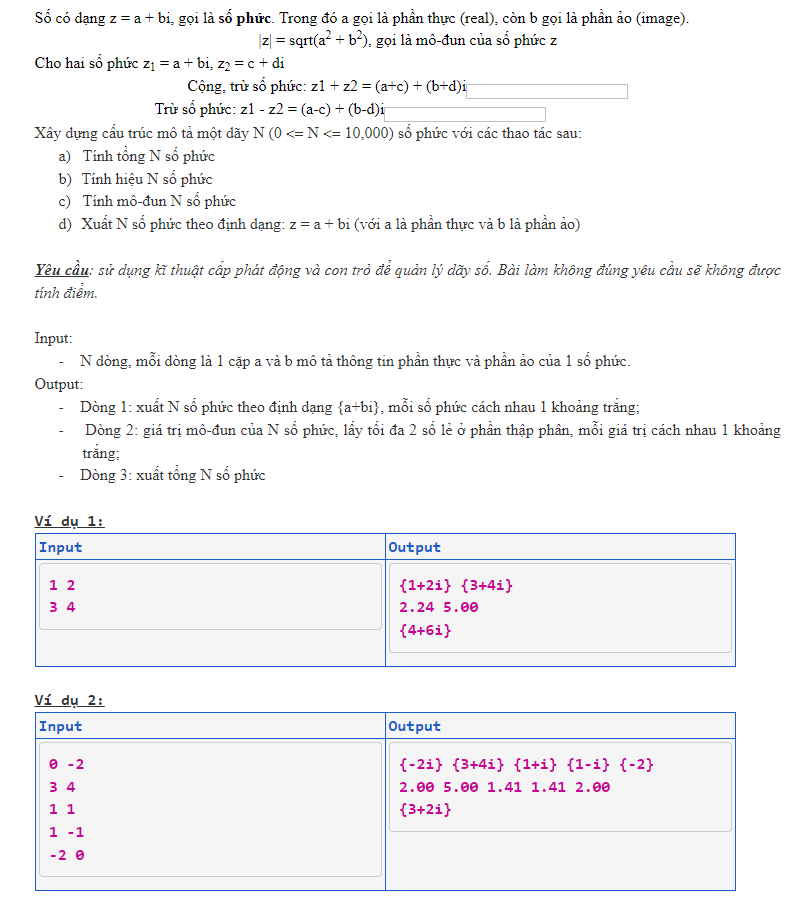
A picture containing graphical user interface

Description automatically generated

CHƯƠNG 3: CON TRỎ

1. 1. 1718\_2\_LTNC\_Midterm\_01

Đề bài



Code

#include <bits/stdc++.h>

using namespace std;

class SoPhuc {

private:

int a, b;

public:

int geta() { return a; }

void seta(int \_a) { a = \_a; }

int getb() { return b; }

void setb(int \_b) { b = \_b; }

SoPhuc(int \_a = 0, int \_b = 0) {

a = \_a;

b = \_b;

}

SoPhuc(const SoPhuc& z) {

a = z.a;

b = z.b;

}

~SoPhuc() {}

friend istream& operator>> (istream& is, SoPhuc& z) {

is >> z.a >> z.b;

return is;

}

friend ostream& operator<< (ostream& os, SoPhuc z) {

os << "{";

if (z.a != 0) {

os << z.a;

if (z.b == 1)

os << "+i}";

else {

if (z.b == -1)

os << "-i}";

else {

if (z.b < 0)

os << "-" << abs(z.b) << "i}";

if (z.b > 0)

os << "+" << abs(z.b) << "i}";

if (z.b == 0)

os << "}";

}

}

}

else {

if (z.b == 1)

os << "i}";

else {

if (z.b == -1)

os << "-i}";

else {

if (z.b < 0)

os << "-" << abs(z.b) << "i}";

if (z.b > 0)

os << abs(z.b) << "i}";

if (z.b == 0)

os << "}";

}

}

}

return os;

}

SoPhuc operator+ (SoPhuc z) {

this->a = this->a + z.a;

this->b = this->b + z.b;

return \*this;

}

SoPhuc operator- (SoPhuc z) {

this->a = this->a - z.a;

this->b = this->b - z.b;

return \*this;

}

float modun() {

float s = a \* a + b \* b;

return sqrt(s);

}

};

long n;

SoPhuc temp[100000], Tong;

SoPhuc\* z;

int main() {

n = 0;

while (cin >> temp[n])

n++;

z = temp;

for (long i = 0; i < n; i++) {

cout << \*(z + i) << " ";

Tong = Tong + \*(z + i);

}

cout << endl;

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

SoPhuc a = \*(z + i);

cout << fixed << setprecision(2) << a.modun() << " ";

}

cout << endl;

cout << Tong;

return 0;

}

Testcase1

Graphical user interface, text, application, Word

Description automatically generated

Testcase2

Graphical user interface, text, application

Description automatically generated with medium confidence

Testcase3

Word

Description automatically generated

* 1. CONTRO1

Đề bài

Text

Description automatically generated

Code

#include <bits/stdc++.h>

using namespace std;

int main() {

int n;

cin >> n;

int\* a = new int[n];

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

cin >> \*(a + i);

}

int\* Max;

Max = a + 1;

int\* Min;

Min = a + 1;

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

if (\*Max < \*(a + i))

Max = (a + i);

if (\*Min > \*(a + i))

Min = (a + i);

}

cout << \*Max << endl;

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

if (\*Max == \*(a + i))

cout << i << " ";

}

cout << endl << \*Min << endl;

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

if (\*Min == \*(a + i))

cout << i << " ";

}

delete[] a;

return 0;

}

Testcase1

Graphical user interface, Word

Description automatically generated with medium confidence

Testcase2

Graphical user interface, application, Word

Description automatically generated

Testcase3

Graphical user interface, application, Word

Description automatically generated

* 1. CONTRO2

Đề bài

Graphical user interface, text, application, email

Description automatically generated

Code

#include <iostream>

using namespace std;

struct Node {

int data;

Node\* next;

};

Node\* CreateNode(int init\_Data) {

Node\* node = new Node;

node->data = init\_Data;

node->next = NULL;

return node;

}

struct List {

Node\* head;

Node\* tail;

};

void CreateList(List& l) {

l.head = NULL;

l.tail = NULL;

}

void Addhead(List& l, Node\* node) {

if (l.head == NULL) {

l.head = node;

l.tail = node;

}

else {

node->next = l.head;

l.head = node;

}

}

void Addtail(List& l, Node\* node) {

if (l.head == NULL) {

l.head = node;

l.tail = node;

}

else {

l.tail->next = node;

l.tail = node;

}

}

void PrintList(List& l) {

if (l.head != NULL) {

Node\* node = l.head;

while (node != NULL) {

cout << node->data << " ";

node = node->next;

}

}

}

int DeleteK(List& l, int k) {

int dem = 0;

Node\* pp = l.head;

Node\* p = l.head->next;

while (p != NULL) {

if (p->data == k) {

pp->next = p->next;

delete p;

dem++;

p = pp->next;

}

else {

pp = p;

p = p->next;

}

}

return dem;

}

int main() {

int n, k, x;

cin >> n >> k;

List l;

CreateList(l);

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

cin >> x;

Addtail(l, CreateNode(x));

}

int dem = DeleteK(l, k);

if (l.head->data == k) {

Node\* p = l.head;

l.head = p->next;

delete p;

dem++;

}

cout << n - dem << endl;

PrintList(l);

return 0;

}

Testcase1

Graphical user interface

Description automatically generated with low confidence

Testcase2

Graphical user interface

Description automatically generated

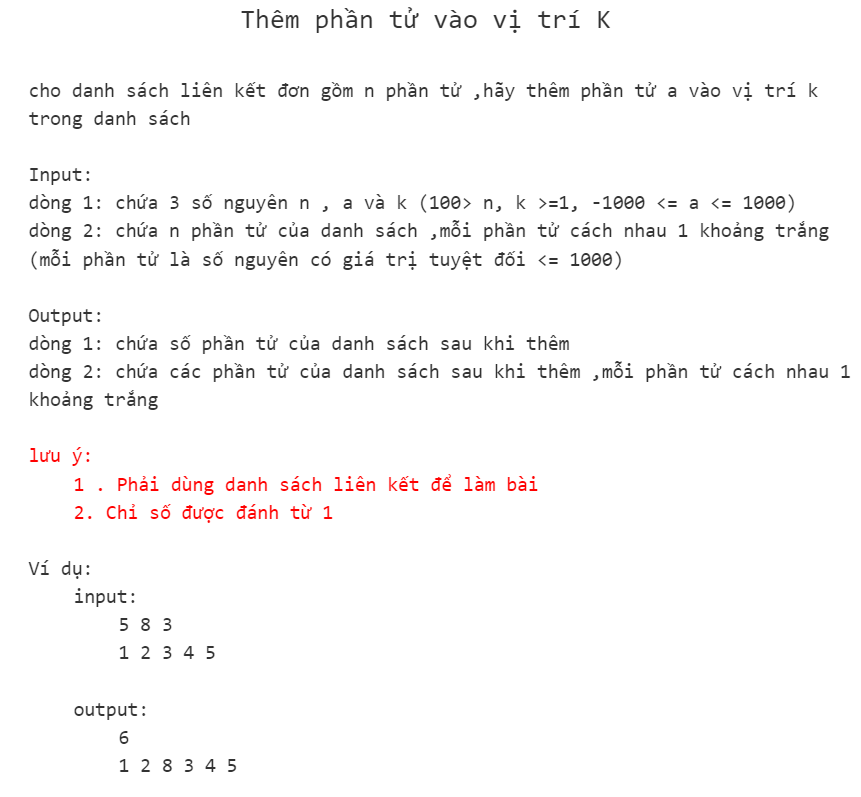
Testcase3

Graphical user interface, application

Description automatically generated

* 1. CONTRO3

Đề bài



Code

#include <bits/stdc++.h>

using namespace std;

struct Node {

int data;

Node\* next;

};

Node\* CreateNode(int init\_data) {

Node\* node = new Node;

node->data = init\_data;

node->next = NULL;

return node;

}

struct List {

Node\* head;

Node\* tail;

};

void CreateList(List& l) {

l.head = l.tail = NULL;

}

void Addhead(List& l, Node\* node) {

if (l.head == NULL)

l.head = l.tail = node;

else {

node->next = l.head;

l.head = node;

}

}

void Addtail(List& l, Node\* node) {

if (l.head == NULL)

l.head = l.tail = node;

else {

l.tail->next = node;

l.tail = node;

}

}

void PrintList(List& l) {

if (l.head != NULL) {

Node\* node = l.head;

while (node != NULL) {

cout << node->data << " ";

node = node->next;

}

}

}

void InsertK(List& l, Node\* a, int k, int& n) {

if (k == 1)

Addhead(l, a);

else {

int i = 1;

Node\* node = l.head;

while (node != NULL) {

if (i == k - 1)

break;

i++;

node = node->next;

}

a->next = node->next;

node->next = a;

}

n++;

}

int main() {

int n, a, k, x;

cin >> n >> a >> k;

List l;

CreateList(l);

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

cin >> x;

Addtail(l, CreateNode(x));

}

InsertK(l, CreateNode(a), k, n);

cout << n << endl;

PrintList(l);

return 0;

}

Testcase1



Testcase2

Text

Description automatically generated with low confidence

Testcase3

Graphical user interface, application

Description automatically generated

* 1. CONTRO5

Đề bài

Graphical user interface, text, application, email

Description automatically generated

Code

#include <bits/stdc++.h>

using namespace std;

struct Node {

int data;

Node\* next;

};

Node\* CreateNode(int init\_data) {

Node\* node = new Node;

node->data = init\_data;

node->next = NULL;

return node;

}

struct List {

Node\* head;

Node\* tail;

List() {

head = tail = NULL;

}

};

void AddTail(List& l, Node\* node) {

if (l.head == NULL)

l.head = l.tail = node;

else {

l.tail->next = node;

l.tail = node;

}

}

int main() {

int n, x;

List l;

cin >> n;

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

cin >> x;

AddTail(l, CreateNode(x));

}

for (Node\* p = l.head; p != NULL; p = p->next) {

for (Node\* pp = l.head; pp != NULL; pp = pp->next) {

for (Node\* ppp = l.head; ppp != NULL; ppp = ppp->next) {

int a = p->data;

int b = pp->data;

int c = ppp->data;

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

if (a == b + c) {

cout << "YES";

return 0;

}

}

}

}

}

cout << "NO";

return 0;

}

Testcase1



Testcase2

A picture containing graphical user interface

Description automatically generated

Testcase3

Graphical user interface

Description automatically generated with low confidence

* 1. Josephus

Đề bài

Text

Description automatically generated with medium confidence

Code

#include <bits/stdc++.h>

using namespace std;

struct Node {

int data;

Node\* next;

};

Node\* CreateNode(int init\_data) {

Node\* node = new Node;

node->data = init\_data;

node->next = NULL;

return node;

}

struct List {

Node\* head;

Node\* tail;

List() {

head = tail = NULL;

}

};

void Addtail(List& l, Node\* node) {

if (l.head == NULL)

l.head = l.tail = node;

else {

l.tail->next = node;

l.tail = node;

}

}

void PrintList(List& l) {

for (Node\* node = l.head; node != NULL; node = node->next)

cout << node->data << " ";

}

int main() {

List l;

int N, M; cin >> N >> M;

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

Addtail(l, CreateNode(i));

Node\* ppp = l.head;

Node\* pp = l.tail;

l.tail->next = l.head;

int dem = 0;

while (N > 0) {

dem++;

if (dem == M) {

N--;

dem = 0;

cout << ppp->data << " ";

pp->next = ppp->next;

delete ppp;

ppp = pp->next;

continue;

}

pp = ppp;

ppp = ppp->next;

}

// PrintList(l);

return 0;

}

Testcase1

Graphical user interface

Description automatically generated with medium confidence

Testcase2

Graphical user interface

Description automatically generated with low confidence

Testcase3

Graphical user interface

Description automatically generated

* 1. Pointer\_Array\_01

Đề bài

Graphical user interface, text, application, email

Description automatically generated

Code

#include <bits/stdc++.h>

using namespace std;

int main() {

int n, j = 0, dem = 0;

cin >> n;

int\* a = new int[n];

int\* b = new int[n];

int\* c = new int[n];

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

cin >> \*(a + i);

}

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

\*(b + j) = \*(a + i);

j++;

}

n /= 2;

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

if (\*(b + i) != \*(a + i)) {

\*(c + dem) = i;

dem++;

}

}

if (dem == 0) {

cout << "mang doi xung";

}

else {

cout << "mang khong doi xung" << endl;

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

cout << \*(a + \*(c + i)) << " " << \*(b + \*(c + i)) << endl;

}

}

return 0;

}

Testcase1

A picture containing text

Description automatically generated

Testcase2

Graphical user interface, application, Word

Description automatically generated

Testcase3

Graphical user interface, application, Word

Description automatically generated

* 1. Pointer\_Mang\_2

Đề bài

Text, email

Description automatically generated

Code

#include <bits/stdc++.h>

using namespace std;

int n;

int\*\* a;

int main() {

cin >> n;

a = new int\* [n];

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

\*(a + i) = new int[n];

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

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

if (i == j)

\*(\*(a + i) + j) = 1;

else

\*(\*(a + i) + j) = 0;

cout << \*(\*(a + i) + j) << " ";

}

cout << endl;

}

return 0;

}

Testcase1

Graphical user interface, application

Description automatically generated

Testcase2

Graphical user interface, application

Description automatically generated

Testcase3

Graphical user interface, application, Word

Description automatically generated

* 1. Pointer\_MissingPrime

Đề bài

Graphical user interface, text, application, email

Description automatically generated

Code

#include <bits/stdc++.h>

using namespace std;

int m, M;

bool\* prime;

int main() {

cin >> m >> M;

prime = new bool[M];

prime[0] = prime[1] = 1;

for (int i = 2; i \* i <= M; i++) {

if (!prime[i]) {

int x = i \* i;

while (x <= M) {

prime[x] = 1;

x += i;

}

}

}

for (int i = m; i <= M; i++)

if (!prime[i])

cout << i << " ";

return 0;

}

Testcase1



Testcase2

Graphical user interface

Description automatically generated with medium confidence

Testcase3

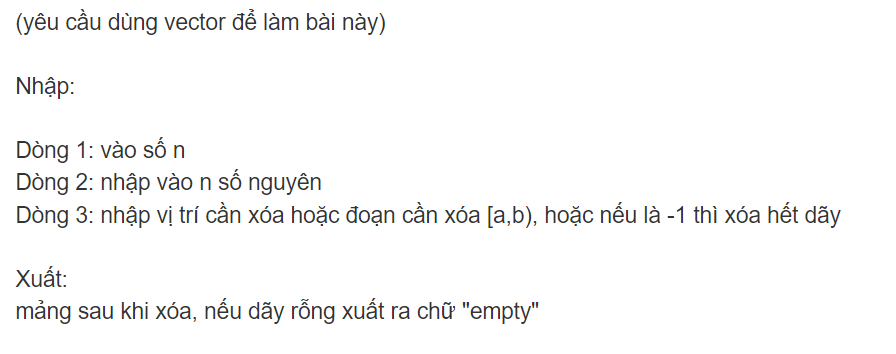
A picture containing graphical user interface

Description automatically generated

CHƯƠNG 4: VECTOR

2. 1. xoa\_vector

Đề bài



Code

#include <bits/stdc++.h>

using namespace std;

int main() {

vector<int> v, k;

int n;

cin >> n;

int x;

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

cin >> x;

v.push\_back(x);

}

while (cin >> x)

k.push\_back(x);

if (k.size() == 1)

if (k[0] == -1)

v.clear();

else {

v.erase(v.begin() + k[0]);

}

else {

v.erase(v.begin() + k[0], v.begin() + k[1]);

}

if (v.empty())

cout << "empty";

else

for (int i = 0 ; i < v.size() ; i++)

cout << v[i] << " ";

return 0;

}

Testcase1

Graphical user interface, application, Word

Description automatically generated

Testcase2

Graphical user interface, application

Description automatically generated

Testcase3

Graphical user interface, application

Description automatically generated

* 1. VECTOR3

Đề bài

Table

Description automatically generated with low confidence

Code

#include <bits/stdc++.h>

using namespace std;

int main() {

vector<int> v;

int x;

while (cin >> x)

if (sqrt(x) != (int)sqrt(x))

v.push\_back(x);

sort(v.begin(), v.end());

for (auto i : v)

if (i%2 == 0)

cout << i << " ";

return 0;

}

Testcase1



Testcase2

Text

Description automatically generated with medium confidence

Testcase3

Graphical user interface, text, application

Description automatically generated

* 1. Vector

Đề bài

Graphical user interface, text, application

Description automatically generated

Code

#include <iostream>

#include <cmath>

#include <vector>

using namespace std;

bool isPrime(long n) {

if (n == 2 || n == 3)

return 1;

if (n%2 == 0)

return 0;

if (n%3 == 0)

return 0;

if (n == 1)

return 0;

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

if (n%i == 0)

return 0;

}

return 1;

}

long sum(long n) {

long s = 0;

while (n > 0) {

if (!isPrime(n%10))

s += n%10;

n /= 10;

}

return s;

}

int main() {

int n;

long x;

vector<long> v;

cin >> n;

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

cin >> x;

v.push\_back(x);

}

for (vector<long>::iterator it = v.begin(); it != v.end(); ++it) {

if (isPrime(\*it))

cout << sum(\*it);

else {

cout << -1;

}

cout << endl;

}

return 0;

}

Testcase1

Graphical user interface, application, Word

Description automatically generated

Testcase2

Graphical user interface

Description automatically generated with medium confidence

Testcase3

Graphical user interface, application, Word

Description automatically generated

* 1. Vector2

Đề bài

Graphical user interface, text, application, email

Description automatically generated

Code

#include <iostream>

#include <string>

#include <vector>

using namespace std;

int main() {

string s;

getline(cin, s);

int x;

vector<int> v;

while (cin >> x)

v.push\_back(x);

if (s.size() == 1)

v.erase(v.begin() + (int(s[0]) - 48));

else {

if (s == "-1") {

v.clear();

cout << "empty";

return 0;

}

else {

v.erase(v.begin() + (int(s[0]) - 48), v.begin() + (int(s[2]) - 48));

}

}

for (vector<int>::iterator it = v.begin(); it != v.end(); ++it) {

cout << \*it << " ";

}

return 0;

}

Testcase1

Graphical user interface, application

Description automatically generated

Testcase2

A picture containing text

Description automatically generated

Testcase3

Graphical user interface

Description automatically generated with medium confidence

* 1. Vector1

Đề bài

Text, letter

Description automatically generated

Code

#include <bits/stdc++.h>

using namespace std;

int main() {

vector<int> v;

int x;

while (cin >> x)

v.push\_back(x);

for (vector<int>::iterator it = v.begin() ; it != v.end() ; it++)

cout << \*it << ' ';

cout << endl;

for (vector<int>::reverse\_iterator it = v.rbegin() ; it != v.rend() ; it++) {

cout << \*it << ' ';

}

return 0;

}

Testcase1

Graphical user interface, application

Description automatically generated with medium confidence

Testcase2

Graphical user interface, application

Description automatically generated with medium confidence

Testcase3

A picture containing text

Description automatically generated

CHƯƠNG 5: NHẬP XUẤT FILE

5. 1. ONTAP

Đề bài

Graphical user interface, text, application, email

Description automatically generated

Text

Description automatically generated with medium confidence

Code

#include <bits/stdc++.h>

using namespace std;

int ucln(int a, int b) {

while (b > 0) {

int r = a%b;

a = b;

b = r;

}

return a;

}

struct PhanSo {

int a, b;

PhanSo() {

a = 0;

b = 1;

}

};

void rutGon(PhanSo& p) {

int x = ucln(p.a, p.b);

p.a /= x;

p.b /= x;

}

istream& operator>> (istream& is, PhanSo& p) {

char x;

is >> p.a >> x >> p.b;

return is;

}

ostream& operator<< (ostream& os, PhanSo p) {

rutGon(p);

if (p.a == 0) {

os << 0;

return os;

}

if (p.b == 1) {

os << p.a;

return os;

}

os << p.a << "/" << p.b;

return os;

}

bool operator== (PhanSo a, PhanSo b) {

return a.a\*b.b == a.b\*b.a;

}

bool operator< (PhanSo a, PhanSo b) {

return a.a\*b.b < a.b\*b.a;

}

bool operator> (PhanSo a, PhanSo b) {

return a.a\*b.b > a.b\*b.a;

}

PhanSo operator+ (PhanSo a, PhanSo b) {

PhanSo c;

c.a = a.a\*b.b + a.b\*b.a;

c.b = a.b\*b.b;

rutGon(c);

return c;

}

template <typename Temp>

struct Mang {

Temp arr[1001];

int n;

Mang() {

n = 0;

}

void operator = (Mang m){

n = m.n;

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

arr[i] = m.arr[i];

}

};

template <typename Temp>

istream& operator>> (istream& is, Mang<Temp>& t) {

is >> t.n;

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

is >> t.arr[i];

return is;

}

template <typename Temp>

ostream& operator<< (ostream& os, Mang<Temp> t) {

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

os << t.arr[i] << " ";

return os;

}

template <typename Temp>

Temp tongPhanTu (Mang<Temp> m) {

Temp t = m.arr[0];

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

t = t + m.arr[i];

return t;

}

template <typename Temp>

Temp phanTuMax (Mang<Temp> m) {

Temp t = m.arr[0];

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

if (t < m.arr[i])

t = m.arr[i];

return t;

}

template <typename Temp>

Mang<Temp> sapXep(Mang<Temp> m, char type = '<') {

if (type == '<') {

for (int i = 0 ; i < m.n ; i++) {

for (int j = i + 1 ; j < m.n ; j++) {

if (m.arr[i] > m.arr[j]) {

Temp t = m.arr[i];

m.arr[i] = m.arr[j];

m.arr[j] = t;

}

}

}

return m;

}

for (int i = 0 ; i < m.n ; i++) {

for (int j = i + 1 ; j < m.n ; j++) {

if (m.arr[i] < m.arr[j]) {

Temp t = m.arr[i];

m.arr[i] = m.arr[j];

m.arr[j] = t;

}

}

}

return m;

}

template <typename Temp>

Mang<Temp> operator+ (Mang<Temp> a, Mang<Temp> b) {

Mang<Temp> c;

if (a.n > b.n) {

c = a;

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

c.arr[i] = c.arr[i] + b.arr[i];

return c;

}

c = b;

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

c.arr[i] = c.arr[i] + a.arr[i];

return c;

}

int main() {

char value;

cin >> value;

if (value == 'I') {

Mang<int> m1, m2;

cin >> m1 >> m2;

cout << m1 << endl;

cout << "MAX 1 = " << phanTuMax(m1) << endl;

cout << "SUM 1 = " << tongPhanTu(m1) << endl;

Mang<int> m3 = sapXep(m1);

for (int i = 0 ; i < m3.n ; i++)

if (m3.arr[i]%2 == 0)

cout << m3.arr[i] << ' ';

for (int i = 0 ; i < m3.n ; i++)

if (m3.arr[i]%2 != 0)

cout << m3.arr[i] << ' ';

cout << endl << m2 << endl;

cout << "MAX 2 = " << phanTuMax(m2) << endl;

cout << "SUM 2 = " << tongPhanTu(m2) << endl;

Mang<int> m4 = sapXep(m2, '>');

for (int i = 0 ; i < m4.n ; i++)

if (m4.arr[i]%2 != 0)

cout << m4.arr[i] << ' ';

for (int i = 0 ; i < m4.n ; i++)

if (m4.arr[i]%2 == 0)

cout << m4.arr[i] << ' ';

cout << endl << m1 + m2;

return 0;

}

Mang<PhanSo> m1, m2;

cin >> m1 >> m2;

cout << m1 << endl;

cout << "MAX 1 = " << phanTuMax(m1) << endl;

cout << "SUM 1 = " << tongPhanTu(m1) << endl;

Mang<PhanSo> m3 = sapXep(m1);

cout << m3 << endl;

cout << m2 << endl;

cout << "MAX 2 = " << phanTuMax(m2) << endl;

cout << "SUM 2 = " << tongPhanTu(m2) << endl;

Mang<PhanSo> m4 = sapXep(m2, '>');

cout << m4 << endl;

cout << m1 + m2;

return 0;

}

Testcase1

Graphical user interface, application

Description automatically generated

Testcase2

Graphical user interface, application

Description automatically generated

Testcase3

Graphical user interface, application

Description automatically generated

* 1. FSTREAM1

Đề bài

Graphical user interface, text, application, email

Description automatically generated

Code

#include<iostream>

#include<fstream>

#include<math.h>

using namespace std;

int main(){

ifstream in("FSTREAM.inp");

int a;

in >> a;

ofstream out("FSTREAM.out");

if((float)sqrt(a) == (int)sqrt(a)){

out << "YES";

}else{

out << "NO";

}

in.close();

out.close();

return 0;

}

Testcase1

Graphical user interface, application

Description automatically generated

Testcase2

Graphical user interface

Description automatically generated with low confidence

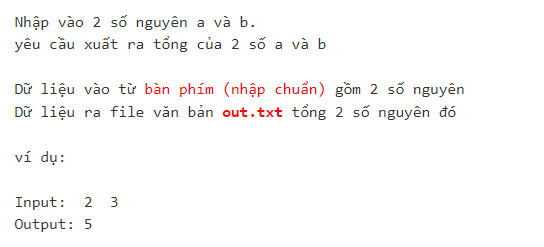
Testcase3

Graphical user interface, application

Description automatically generated

* 1. fstream3

Đề bài



Code

#include <bits/stdc++.h>

using namespace std;

int main() {

int a, b;

cin >> a >> b;

ofstream out("out.txt");

out << a + b;

out.close();

return 0;

}

Testcase1

A picture containing text

Description automatically generated

Testcase2

Graphical user interface, application

Description automatically generated

Testcase3

Graphical user interface

Description automatically generated with medium confidence

* 1. fstream2

Đề bài

Text

Description automatically generated

Code

#include<iostream>

#include<fstream>

using namespace std;

int main(){

ifstream in("input.txt");

int a,b;

in >> a >> b;

ofstream out("output.txt");

out << a + b;

in.close();

out.close();

return 0;

}

Testcase1

A picture containing graphical user interface

Description automatically generated

Testcase2

A picture containing graphical user interface

Description automatically generated

Testcase3

Graphical user interface, application, chat or text message

Description automatically generated

* 1. fstream1

Đề bài

Text

Description automatically generated

Code

#include <iostream>

#include <fstream>

using namespace std;

int main(){

ifstream in("input.txt");

int a,b;

in >> a >> b;

cout << a + b;

in.close();

return 0;

}

Testcase1

A picture containing shape

Description automatically generated

Testcase2

Graphical user interface, application

Description automatically generated

Testcase3

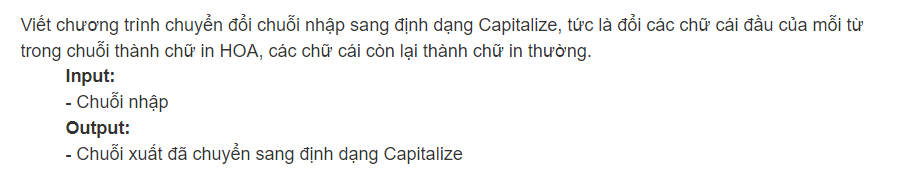
Graphical user interface, application, chat or text message

Description automatically generated

CHƯƠNG 6: STRING (CHUỖI)

6. 1. Capitalize

Đề bài



Code

#include <bits/stdc++.h>

using namespace std;

string s, token;

stringstream ss;

int main() {

getline(cin, s);

ss << s;

while (ss >> token) {

if (token[0] >= 'a' && token[0] <= 'z')

token[0] = char(int(token[0]) - 32);

for (int i = 1; i < token.size(); i++) {

if (token[i] >= 'A' && token[i] <= 'Z')

token[i] = char(int(token[i]) + 32);

}

cout << token << " ";

}

return 0;

}

Testcase1

Application

Description automatically generated with low confidence

Testcase2

Graphical user interface, application

Description automatically generated

Testcase3

Graphical user interface, text, application

Description automatically generated

* 1. CHUSOTANCUNG

Đề bài

Text

Description automatically generated

Code

#include <iostream>

#include <string>

using namespace std;

int main() {

int a, n;

cin >> a >> n;

string str\_a = to\_string(a); // chuyển a thành chuỗi

string res = "1";

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

string temp = "";

int carry = 0;

for (int j = str\_a.size() - 1; j >= 0; j--) {

int num = (str\_a[j] - '0') \* (res[res.size() - 1] - '0') + carry;

temp = char(num % 10 + '0') + temp;

carry = num / 10;

}

if (carry > 0) {

temp = char(carry + '0') + temp;

}

res = temp;

}

cout << res[res.size() - 1] << endl; // lấy ký tự cuối cùng của chuỗi res

return 0;

}

Testcase1

A picture containing graphical user interface

Description automatically generated

Testcase2

Graphical user interface, application

Description automatically generated

Testcase3

Graphical user interface, application

Description automatically generated with medium confidence

* 1. DAONGUOC

Đề bài

A picture containing text

Description automatically generated

Code

#include <iostream>

using namespace std;

int main()

{

string a;

cin >> a;

reverse(a.begin(), a.end());

cout << a;

return 0;

}

Testcase1

Graphical user interface

Description automatically generated with low confidence

Testcase2

Graphical user interface, application

Description automatically generated with medium confidence

Testcase3

A picture containing text

Description automatically generated

* 1. DAONGUOC1

Đề bài



Code

#include<iostream>

#include<string>

using namespace std;

int reverse(int a) {

int result = 0;

while (a > 0) {

result = result \* 10 + a % 10;

a /= 10;

}

return result;

}

int main() {

int a[100];

int n = 0;

while(cin >> a[n++]){}

n--;

int max = a[0];

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

//reverse(a[i]);

if (max < reverse(a[i])) {

max = reverse(a[i]);

}

}

cout << reverse(max);

return 0;

}

Testcase1

Graphical user interface, application

Description automatically generated

Testcase2

Graphical user interface, application

Description automatically generated

Testcase3

Graphical user interface

Description automatically generated

* 1. DAONGUOC2

Đề bài

Text

Description automatically generated

Code

#include<iostream>

#include<string>

#include<fstream>

using namespace std;

int reverse(int a) {

int result = 0;

while (a > 0) {

result = result \* 10 + a % 10;

a /= 10;

}

return result;

}

int main() {

ifstream in("inDaoNguoc2.txt");

int a[100];

int n = 0;

while (in >> a[n++]) {}

n--;

int max = a[0];

int count = 0;

int\* index = new int[n];

ofstream out("outDaoNguoc2.txt");

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

if (max < reverse(a[i])) {

max = reverse(a[i]);

count = 0;

index[count++] = i;

} else if(reverse(a[i]) == max)

index[count++] = i;

}

out << reverse(max) << endl;

if (count > 1) {

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

out << index[i] << " ";

}

}

delete[] index;

in.close();

out.close();

return 0;

}

Testcase1

A picture containing application

Description automatically generated

Testcase2

Graphical user interface, application

Description automatically generated

Testcase3

Graphical user interface, application

Description automatically generated

* 1. DAONGUOC3

Đề bài

Text

Description automatically generated

Code

#include <iostream>

#include <string>

#include <algorithm>

#include <sstream>

using namespace std;

int main() {

string a;

getline(cin, a);

string b[1000];

int n = 0;

stringstream ss(a);

while(ss>>b[n++]){}

n--;

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

cout << b[i] << " ";

}

}

Testcase1

A picture containing shape

Description automatically generated

Testcase2

A picture containing text

Description automatically generated

Testcase3

A picture containing graphical user interface

Description automatically generated

* 1. DEMCHUOI

Đề bài

Graphical user interface, text

Description automatically generated with medium confidence

Code

#include<iostream>

#include<string>

using namespace std;

int main()

{

int n;

cin >> n;

cin.ignore();

string s\_tim, s\_nhap;

getline(cin, s\_tim);

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

int count = 0;

getline(cin, s\_nhap);

size\_t pos = 0;

while ((pos = s\_nhap.find(s\_tim, pos)) != string::npos) {

count++;

pos += s\_nhap.find(s\_tim) + 1;

}

// cout << s\_tim<<endl;

// cout << s\_nhap<<endl;

cout << count << endl;

}

return 0;

}

Testcase1

Graphical user interface, application, Word

Description automatically generated

Testcase2



Testcase3

Graphical user interface, application

Description automatically generated

* 1. DOIXUNG

Đề bài

Text, letter

Description automatically generated

Code

#include<iostream>

#include<string>

#include<algorithm>

using namespace std;

bool is\_check(string s) {

s.erase(remove(s.begin(), s.end(), ' '), s.end());

transform(s.begin(), s.end(), s.begin(), ::tolower);

/\*for (int i = 0; i < s.length(); i++) {

if (s[i] == ' ') {

s.erase(s.begin() + i, s.begin() + i + 1);

}

}\*/

for (int i = 0; i < (s.length() / 2); i++) {

if (s[i] != s[s.length() - i - 1]) {

return false;

}

}

return true;

}

int main()

{

int n;

cin >> n;

cin.ignore();

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

string s;

getline(cin, s);

cout << is\_check(s)<<endl;

}

return 0;

}

Testcase1

Graphical user interface, application, Word

Description automatically generated

Testcase2

Graphical user interface, application

Description automatically generated

Testcase3

Graphical user interface, application, Word

Description automatically generated

* 1. DATE

Đề bài

Text, letter

Description automatically generated

Text, letter

Description automatically generated

Code

#include <iostream>

#include <string>

using namespace std;

struct Date

{

int month, day, year;

};

istream& operator>>(istream& inp, Date& d)

{

inp >> d.day >> d.month >> d.year;

return inp;

}

ostream& operator<<(ostream& out, Date d)

{

if (d.day < 10) out << 0 << d.day;

else out << d.day;

out << "/";

if (d.month < 10) out << 0 << d.month;

else out << d.month;

out << "/";

out << d.year;

return out;

}

bool LeapYear(Date d)

{

return (d.year % 4 == 0 && d.year % 100 != 0) || (d.year % 400 == 0);

}

int ndayofYear(Date d)

{

int x = 0, i;

for (i = 1; i < d.month; i++)

{

switch (i)

{

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

{

x += 31;

break;

}

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

{

x += 30;

break;

}

case 2:

{

if (LeapYear(d))

{

x += 29;

}

else

{

x += 28;

}

break;

}

}

}

return x + d.day;

}

string ndayofWeek(Date d)

{

string s;

int n = d.year;

int c = ndayofYear(d);

int t1 = (n - 1) / 4;

int t2 = (n - 1) / 100;

int t3 = (n - 1) / 400;

int k = (n - 1) + t1 - t2 + t3 + c;

switch (k % 7)

{

case 0:

{

s = "Sunday";

break;

}

case 1:

{

s = "Monday";

break;

}

case 2:

{

s = "Tuesday";

break;

}

case 3:

{

s = "Wednesday";

break;

}

case 4:

{

s = "Thursday";

break;

}

case 5:

{

s = "Friday";

break;

}

case 6:

{

s = "Saturday";

break;

}

}

return s;

}

int ndayofMonth(Date d)

{

int nday;

switch (d.month)

{

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

{

nday = 31;

break;

}

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

{

nday = 30;

break;

}

case 2:

{

if (LeapYear(d))

{

nday = 29;

}

else

{

nday = 28;

}

break;

}

}

return nday;

}

Date nextDay(Date d)

{

Date nextd;

if (d.day == ndayofMonth(d))

{

if (d.month == 12)

{

nextd.day = 1;

nextd.month = 1;

nextd.year = d.year + 1;

}

else

{

nextd.day = 1;

nextd.month = d.month + 1;

nextd.year = d.year;

}

}

else

{

nextd.day = d.day + 1;

nextd.month = d.month;

nextd.year = d.year;

}

return nextd;

}

bool operator==(Date d1, Date d2)

{

return d1.day == d2.day && d1.month == d2.month && d1.year == d2.year;

}

void likendayofWeek(Date d1, Date d2)

{

ndayofWeek(d1) == ndayofWeek(d2) ? cout << "TRUE" : cout << "FALSE";

}

bool operator<(Date d1, Date d2)

{

if (d1.year < d2.year)

{

return true;

}

else if (d1.year == d2.year)

{

if (d1.month < d2.month)

{

return true;

}

else if (d1.month == d2.month)

{

if (d1.day < d2.day) return true;

else return false;

}

else return false;

}

else return false;

}

int operator-(Date d1, Date d2)

{

int y;

if (d1.year == d2.year)

{

if (d1 < d2)

{

return ndayofYear(d2) - ndayofYear(d1);

}

else if (d1 == d2) return 0;

else

{

return ndayofYear(d1) - ndayofYear(d2);

}

}

if (d1.year < d2.year)

{

if (LeapYear(d1))

{

y = 366 - ndayofYear(d1);

}

else y = 365 - ndayofYear(d1);

for (int i = d1.year + 1; i < d2.year; i++)

{

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

{

y += 366;

}

else y += 365;

}

return y + ndayofYear(d2);

}

if (d1.year > d2.year)

{

if (LeapYear(d2))

{

y = 366 - ndayofYear(d2);

}

else y = 365 - ndayofYear(d2);

for (int i = d2.year + 1; i < d1.year; i++)

{

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

{

y += 366;

}

else y += 365;

}

return y + ndayofYear(d1);

}

}

int main()

{

Date x1, x2;

cin >> x1;

cin >> x2;

cout << x1 << " " << ndayofWeek(x1) << " " << ndayofYear(x1) << " " << nextDay(x1) << " ";

LeapYear(x1) == true ? cout << "TRUE" : cout << "FALSE";

cout << endl;

cout << x2 << " " << ndayofWeek(x2) << " " << ndayofYear(x2) << " " << nextDay(x2) << " ";

LeapYear(x2) == true ? cout << "TRUE" : cout << "FALSE";

cout << endl;

likendayofWeek(x1, x2);

cout << endl;

if (x1 < x2) cout << "1 < 2";

else if (x1 == x2) cout << "1 = 2";

else cout << "1 > 2";

cout << endl;

cout << x1 - x2;

}

Testcase1

Graphical user interface, text, application

Description automatically generated

Testcase2

Graphical user interface, text, application

Description automatically generated

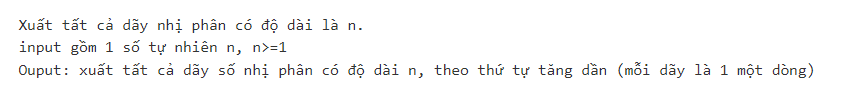
Testcase3

Graphical user interface, text, application, Word

Description automatically generated

* 1. DAYNP

Đề bài



Code

#include <iostream>

#include <string>

#include <math.h>

using namespace std;

string sinhNhiPhan(int so, int doDai)

{

string res = "";

int phanDu;

while(so > 0)

{

phanDu = so % 2;

res = (char)(phanDu + '0') + res;

so /= 2;

}

while(res.size() < doDai)

{

res = '0' + res;

}

return res;

}

void sinhNhiPhanTiepTheo(string& s)

{

int pos = -1;

for(int i = s.size() - 1; i >= 0; i--)

{

if(s[i] == '0')

{

pos = i;

break;

}

}

if(pos == -1) return;

s[pos] = '1';

for(int i = pos + 1; i < s.size(); i++) s[i] = '0';

return;

}

int main()

{

int n;

cin >> n;

int soLuong = pow(2, n);

int cach = 2;

string s = "0";

while(s.size() < n) s += '0';

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

{

if(cach == 1)cout << sinhNhiPhan(i, n) << endl;

else

{

cout << s << endl;

sinhNhiPhanTiepTheo(s);

}

}

return 0;

}

Testcase1

Graphical user interface, application

Description automatically generated

Testcase2

Graphical user interface, application, Word

Description automatically generated

Testcase3

Graphical user interface, application

Description automatically generated

* 1. SoPalindrom

Đề bài

Text

Description automatically generated

Code

#include <iostream>

using namespace std;

int reverse(int a) {

int result = 0;

while (a > 0) {

result = result \* 10 + a % 10;

a /= 10;

}

return result;

}

int main()

{

int n;

int a[1000];

cin >> n;

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

cin >> a[i];

}

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

if (reverse(a[i]) == a[i]) {

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

}

}

return 0;

}

Testcase1

Graphical user interface, application, Word

Description automatically generated

Testcase2

A picture containing graphical user interface

Description automatically generated

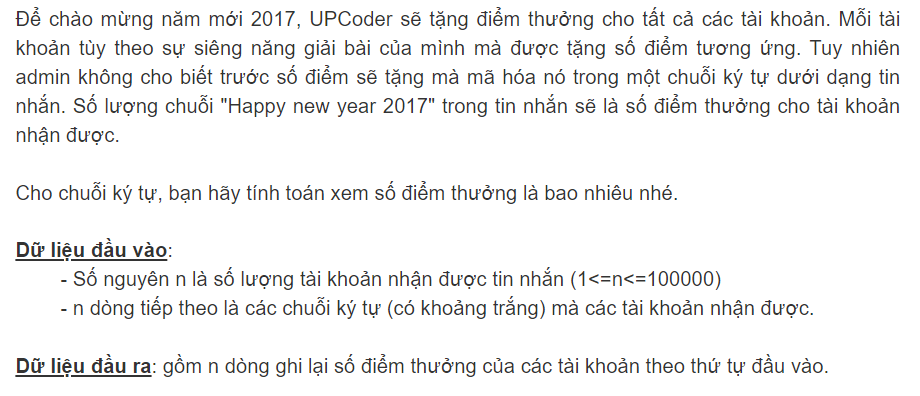
Testcase3

Graphical user interface

Description automatically generated with low confidence

* 1. HAPPYNEWYEAR

Đề bài



Code

#include <iostream>

#include <string>

using namespace std;

string s;

int n, t, dem;

int main() {

cin >> n;

cin.ignore();

while (n > 0) {

getline(cin, s);

dem = 0;

t = s.find("Happy new year 2017");

while (t != -1) {

dem++;

t = s.find("Happy new year 2017", t + 1);

}

cout << dem;

if (n > 1)

cout << endl;

n--;

}

return 0;

}

Testcase1

Graphical user interface, application, Word

Description automatically generated

Testcase2

Graphical user interface, application, Word

Description automatically generated

Testcase3

Graphical user interface, application, Word

Description automatically generated

* 1. Hpalindrome

Đề bài

Text

Description automatically generated

Code

#include<iostream>

#include<fstream>

#include<string>

#include<algorithm>

using namespace std;

bool check(string s) {

s.erase(remove(s.begin(), s.end(), ' '), s.end());

transform(s.begin(), s.end(), s.begin(), ::tolower);

int m = s.length();

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

if (s[i] != s[m - i - 1]) {

return false;

}

}

return true;

}

int main()

{

ifstream in("input.txt");

ofstream out("output.txt");

string s;

while (getline(in, s)) {

if (check(s)) {

out << "1" << endl;

}

else

out << "0" << endl;

}

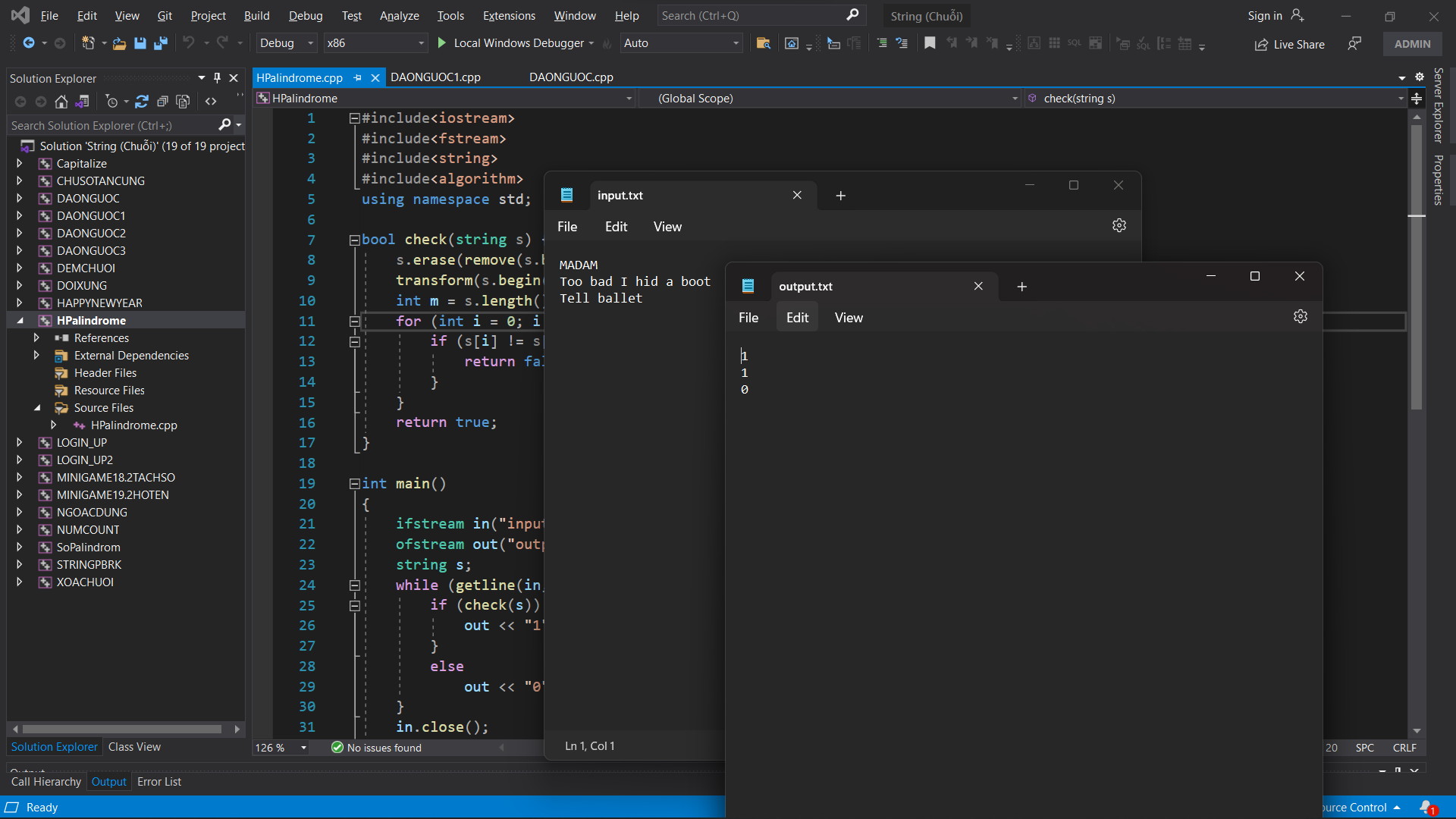
in.close();

out.close();

return 0;

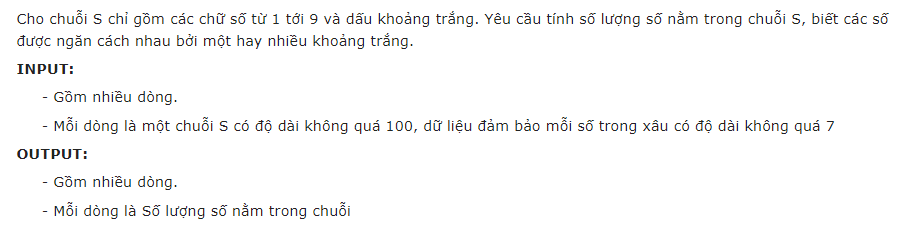
}

Testcase1



* 1. NUMCOUNT

Đề bài



Code

#include <iostream>

#include<string>

using namespace std;

void demso(string& s)

{

int dem = 0;

for (int i = 0; i < s.length(); i++)

{

if (s[i] != ' ' && s[i + 1] == ' ')

{

dem++;

}

}

if (s.size() - 1 != ' ')

{

dem++;

}

cout << dem << "\n";

}

int main()

{

string s;

while (getline(cin, s))

{

demso(s);

}

return 0;

}

Testcase1

Graphical user interface, application

Description automatically generated

Testcase2

Graphical user interface, application

Description automatically generated

Testcase3

Graphical user interface

Description automatically generated with medium confidence

* 1. MINIGAME19.2HOTEN

Đề bài

Text

Description automatically generated

Code

#include <iostream>

#include <string>

using namespace std;

void hoTen(string s[], int i) {

cout << "Ho: " << s[0] << endl;

if (i > 2) {

cout << "Lot: ";

for (int j = 1; j < i - 1; j++)

cout << s[j] << " ";

cout << endl;

}

cout << "Ten: " << s[i - 1];

}

int main() {

string s[50];

int i = 0;

while (cin >> s[i])

i++;

hoTen(s, i);

return 0;

}

Testcase1

Graphical user interface, application, Word

Description automatically generated

Testcase2

Graphical user interface, application, Word

Description automatically generated

Testcase3

Graphical user interface, application, Word

Description automatically generated

* 1. MINIGAME18.2TACHSO

Đề bài

Text

Description automatically generated

Code

#include <sstream>

#include <vector>

#include <iostream>

#include <fstream>

using namespace std;

vector<int> toInt(string str) {

vector<int> v;

stringstream ss;

string tmp = "";

int x;

for (int i = 0; i < str.size(); i++) {

if (str[i] != ',')

tmp = tmp + str[i];

if (str[i] == ',' || i == str.size() - 1) {

//xuat ra cac chu so

ss << tmp;

ss >> x;

v.push\_back(x);

ss.clear();

tmp = "";

}

}

return v;

}

int main() {

string str;

cin >> str;

vector<int> v = toInt(str);

long long sum = 0;

for (int i = 0; i < v.size(); i++) {

cout << v[i] << endl;

sum += v[i];

}

cout << sum << endl;

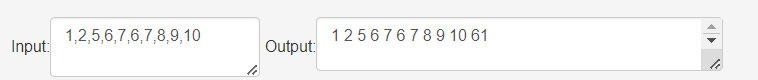
return 0;

}

Testcase1



Testcase2



Testcase3

Graphical user interface, application

Description automatically generated

* 1. LOGIN\_UP

Đề bài

Text

Description automatically generated

Code

#include <bits/stdc++.h>

using namespace std;

int main() {

char a;

int i = 0, dem = 0;

string s = "upcoder";

while (cin >> a) {

if (a == s[i]) {

i++;

dem++;

}

}

if (dem == 7)

cout << "YES";

else

cout << "NO";

return 0;

}

Testcase1

Graphical user interface, application

Description automatically generated

Testcase2

A picture containing graphical user interface

Description automatically generated

Testcase3

Graphical user interface, text

Description automatically generated with medium confidence

* 1. LOGIN\_UP2

Đề bài

Text

Description automatically generated

Code

#include <iostream>

#include <cstring>

#include <stdio.h>

using namespace std;

char a[501][101], demo[] = "upcoder";

int n, m = 0, dd[501];

int check(char a[])

{

int j = 0;

for (int i = 0; i < strlen(a); i++)

if (a[i] == demo[j])

{

j++;

if (j == 7)

return 1;

}

return 0;

}

int main()

{

cin >> n;

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

{

cin >> a[i];

dd[i] = 0;

}

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

if (check(a[i]))

{

dd[i] = 1;

m++;

}

cout << m << endl;

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

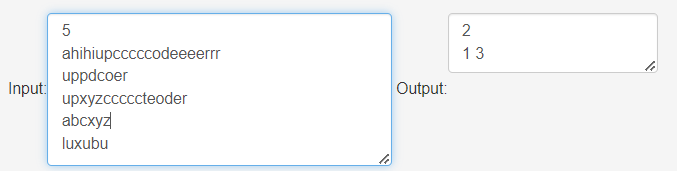
if (dd[i])

cout << i + 1 << " ";

return 0;

}

Testcase1



Testcase2

Graphical user interface, application, Word

Description automatically generated

* 1. STRINGPBRK

Đề bài

Text

Description automatically generated

Code

#include <iostream>

#include <string>

using namespace std;

bool Chars(string s) {

// Kiểm tra xem từ s có chứa ký tự b, d, f, h, k, t hay không.

for (int i = 0; i < s.length(); i++) {

if (s[i] == 'b' || s[i] == 'd' || s[i] == 'f' || s[i] == 'h' || s[i] == 'k' || s[i] == 't') {

return true;

}

}

return false;

}

int main() {

string s, longest = "";

while (cin >> s) {

string word = "";

for (int i = 0; i < s.length(); i++) {

if (s[i] != ' ') {

word += s[i];

}

else {

// Kiểm tra từ hiện tại

if (!Chars(word)) {

// Nếu từ hiện tại không chứa ký tự b, d, f, h, k, t

if (word.length() > longest.length()) {

// Nếu từ hiện tại dài hơn từ dài nhất hiện tại

longest = word;

}

}

word = "";

}

}

// Kiểm tra từ cuối cùng trong chuỗi

if (!Chars(word)) {

if (word.length() > longest.length()) {

longest = word;

}

}

}

cout << longest << endl;

return 0;

}

Testcase1

Graphical user interface, application, Word

Description automatically generated

Testcase2

Graphical user interface, application

Description automatically generated

Testcase3

Graphical user interface, application

Description automatically generated

* 1. XOACHUOI

Đề bài

Text

Description automatically generated

Code

#include<iostream>

#include<string>

using namespace std;

int main()

{

string X, Y;

getline(cin, X);

getline(cin, Y);

//X.erase(remove(X.begin(), X.end(), ' '), X.end());

//Y.erase(remove(Y.begin(), Y.end(), ' '), Y.end());

int i = 0, j = 0, count = 0;

while (i < X.length() && j < Y.length()) {

if (X[i] == Y[j]) {

i++;

j++;

}

else {

j++;

count++;

}

}

if (i == X.length()) {

cout << "YES" << endl;

cout << count << endl;

}

else {

cout << "NO" << endl;

}

return 0;

}

Testcase1

A picture containing graphical user interface

Description automatically generated

Testcase2

A picture containing graphical user interface

Description automatically generated

Testcase3

Graphical user interface, application

Description automatically generated

* 1. 1718\_2\_LTNC\_Midterm\_02

Đề bài

Graphical user interface, text, application, email

Description automatically generated

Code

#include <iostream>

#include <algorithm>

#include <string>

using namespace std;

int main() {

string s1, s2;

cin >> s1 >> s2;

// Kiểm tra độ dài của hai chuỗi

if (s1.length() != s2.length()) {

cout << "NO" << endl;

} else {

// Sắp xếp các ký tự trong hai chuỗi

sort(s1.begin(), s1.end());

sort(s2.begin(), s2.end());

// So sánh hai chuỗi đã sắp xếp

if (s1 == s2) {

cout << "YES" << endl;

} else {

cout << "NO" << endl;

}

}

return 0;

}

Testcase1

Graphical user interface

Description automatically generated with low confidence

Testcase2

A picture containing graphical user interface

Description automatically generated

Testcase3

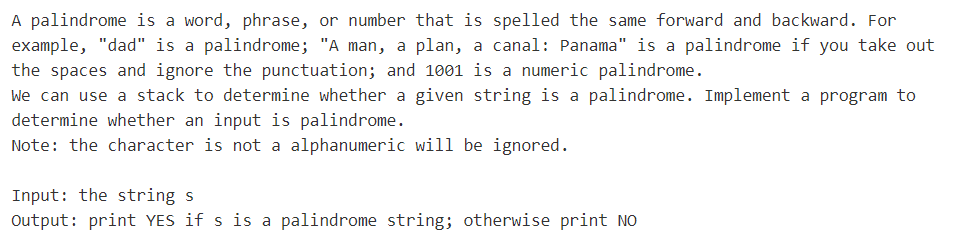
A picture containing application

Description automatically generated

CHƯƠNG 7: STACK VÀ QUEUE

1. 1. PALINDROME\_STRING

Đề bài



Code

#include <bits/stdc++.h>

using namespace std;

string s, x = "";

string t;

int main() {

getline(cin, s);

for (int i = 0 ; i < s.size() ; i++) {

s[i] = tolower(s[i]);

if ((s[i] >= 'a' && s[i] <= 'z') || (s[i] >= '0' && s[i] <= '9'))

x += s[i];

}

t = x;

reverse(t.begin(), t.end());

if (t == x)

cout << "YES";

else

cout << "NO";

return 0;

}

Testcase1

Graphical user interface, text, application, chat or text message

Description automatically generated

Testcase2

Graphical user interface, text, application

Description automatically generated

Testcase3

Graphical user interface, text, application, chat or text message

Description automatically generated

* 1. TRUNGHAUTO2

Đề bài

Graphical user interface, text, application, email

Description automatically generated

Code

#include <bits/stdc++.h>

using namespace std;

int prior(string x) {

if (x == "+" || x == "-")

return 1;

if (x == "\*" || x == "/" || x == "%" || x == "^")

return 2;

}

int main() {

stack<string> st;

string s;

while (cin >> s) {

if (s >= "0" && s <= "9")

cout << s << " ";

else if (s == "(")

st.push(s);

else if (s == ")") {

while (st.top() != "(") {

cout << st.top() << " ";

st.pop();

}

if (st.top() == "(")

st.pop();

}

else {

while (!st.empty() && st.top() != "(" && prior(st.top()) >= prior(s)) {

cout << st.top() << " ";

st.pop();

}

st.push(s);

}

}

while (!st.empty()) {

cout << st.top() << " ";

st.pop();

}

return 0;

}

Testcase1

Graphical user interface, text, application, chat or text message

Description automatically generated

Testcase2

Text

Description automatically generated

Testcase3

A picture containing graphical user interface

Description automatically generated

* 1. TRUNGHAUTO1

Đề bài

Graphical user interface, text, application

Description automatically generated

Code

#include <bits/stdc++.h>

using namespace std;

struct Stack{};

void init();

void push(Stack &st, int x);

int top(Stack st);

void pop(Stack &st);

bool empty(Stack st);

bool full(Stack st);

int prior(char x) {

return (x == '+' || x == '-') ? 1 :

(x == '\*' || x == '/' || x == '%') ? 2 : 0;

}

string s;

stack<char> st;

int main() {

cin >> s;

for (int i = 0 ; i < s.size() ; i++) {

if (isdigit(s[i]))

cout << s[i] << " ";

else if (s[i] == '+' || s[i] == '-' || s[i] == '\*' || s[i] == '/' || s[i] == '%') {

while (!st.empty() && st.top() != '(' && prior(st.top()) >= prior(s[i])) {

cout << st.top() << " ";

st.pop();

}

st.push(s[i]);

}

else if (s[i] == '(')

st.push(s[i]);

else {

while (st.top() != '(') {

cout << st.top() << " ";

st.pop();

}

if (st.top() == '(')

st.pop();

}

}

while (!st.empty()) {

cout << st.top() << " ";

st.pop();

}

return 0;

}

Testcase1

Text

Description automatically generated

Testcase2

Graphical user interface, text, chat or text message

Description automatically generated

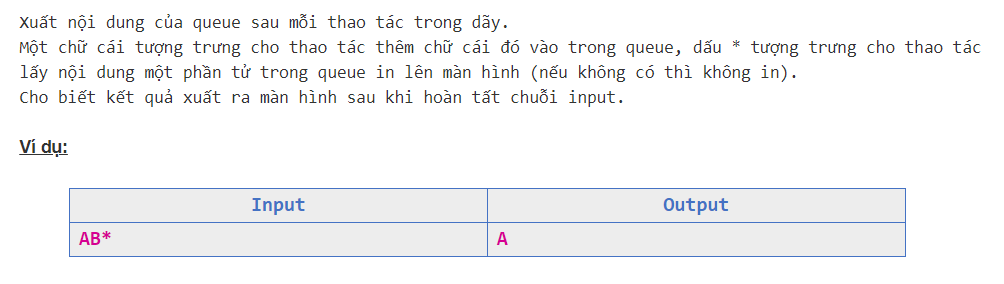
Testcase3

Graphical user interface, application

Description automatically generated

* 1. QUEUE1

Đề bài



Code

#include <bits/stdc++.h>

using namespace std;

queue<char> q;

char x;

int main() {

while (cin >> x) {

if (x == '\*') {

if (!q.empty()) {

cout << q.front();

q.pop();

}

}

else

q.push(x);

}

return 0;

}

Testcase1

Graphical user interface, text, application

Description automatically generated

Testcase2

Graphical user interface, text, application, chat or text message

Description automatically generated

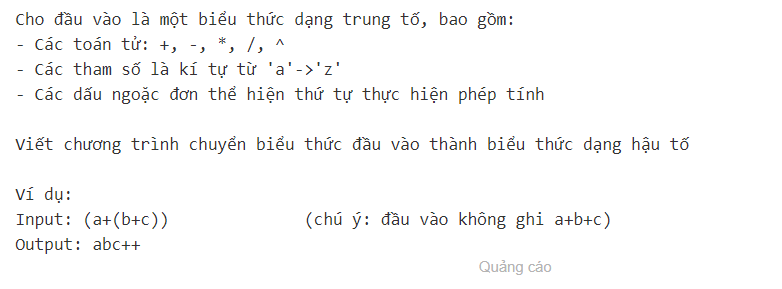
Testcase3

Graphical user interface, text, application, chat or text message

Description automatically generated

* 1. TT\_HT1

Đề bài



Code

#include <bits/stdc++.h>

std::string s;

std::stack<char> st;

int prior(char x) {

return (x == '+' || x == '-') ? 1

: (x == '\*' || x == '/' || x == '^') ? 2 : 0;

}

int main() {

std::cin >> s;

for (int i = 0 ; i < s.size() ; i++) {

if (s[i] >= 'a' && s[i] <= 'z')

std::cout << s[i];

else if (s[i] == '(')

st.push(s[i]);

else if (s[i] == '+' || s[i] == '-' || s[i] == '\*'

|| s[i] == '/' || s[i] == '%' || s[i] == '^') {

while (!st.empty() && st.top() != '(' && prior(st.top()) >= prior(s[i])) {

std::cout << st.top();

st.pop();

}

st.push(s[i]);

}

else {

while (st.top() != '(') {

std::cout << st.top();

st.pop();

}

if (st.top() == '(')

st.pop();

}

}

while (!st.empty()) {

std::cout << st.top();

st.pop();

}

return 0;

}

Testcase1

Graphical user interface, application

Description automatically generated

Testcase2

Graphical user interface, text, application

Description automatically generated

Testcase3

Graphical user interface, text, application, chat or text message

Description automatically generated

* 1. DOICS10\_2

Đề bài

Text

Description automatically generated

Code

#include <bits/stdc++.h>

using namespace std;

int n;

stack<int> s;

int main() {

cin >> n;

if (n == 0) {

cout << 0;

return 0;

}

while (n > 0) {

s.push(n%2);

n = n/2;

}

while (!s.empty()) {

cout << s.top();

s.pop();

}

return 0;

}

Testcase1

Graphical user interface, text, application, chat or text message

Description automatically generated

Testcase2

Graphical user interface, text, application, chat or text message

Description automatically generated

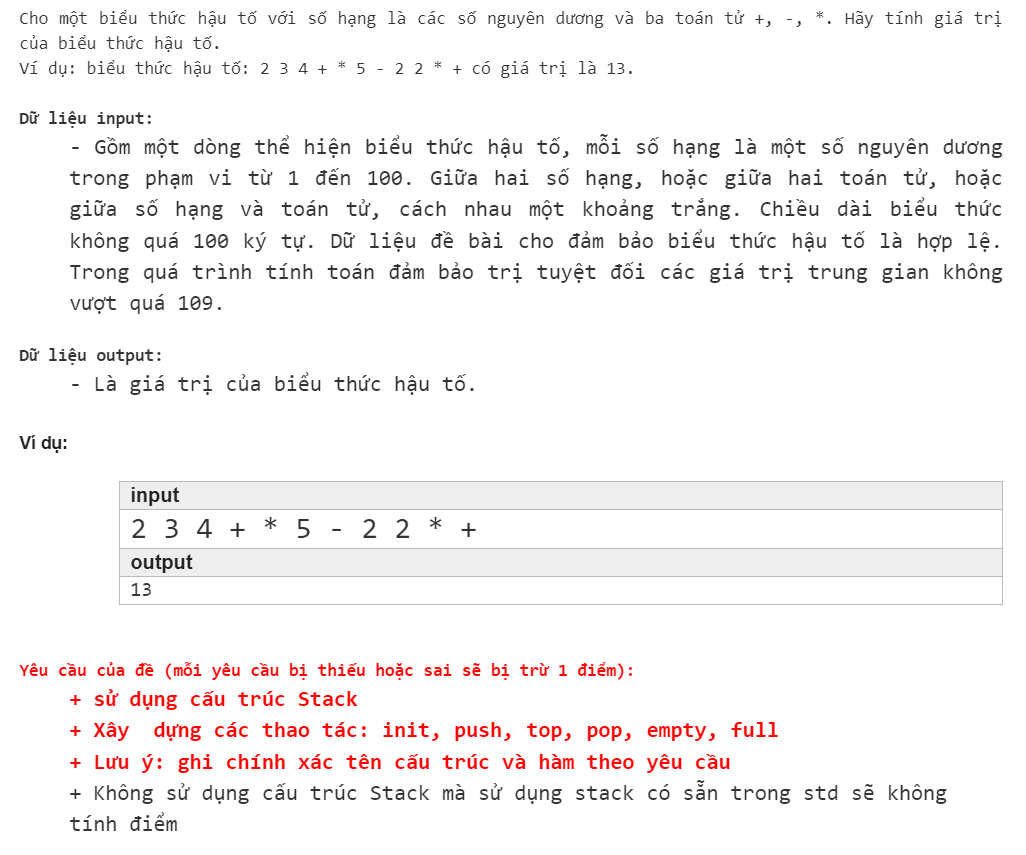
Testcase3

Text

Description automatically generated with medium confidence

* 1. HAUTO

Đề bài



Code

#include <bits/stdc++.h>

using namespace std;

int Stack();

int init ();

int top();

int pop();

int empty();

int full();

string s;

stack<int> st;

int main() {

while (cin >> s) {

if (s == "+" || s == "-" || s == "\*") {

if (!st.empty()) {

int b = st.top(); st.pop();

int a = st.top(); st.pop();

if (s == "+") st.push(a + b);

if (s == "-") st.push(a - b);

if (s == "\*") st.push(a \* b);

}

}

else st.push(stoi(s));

}

cout << st.top();

return 0;

}

Testcase1

Graphical user interface, application

Description automatically generated

Testcase2

A picture containing graphical user interface

Description automatically generated

Testcase3

Graphical user interface

Description automatically generated with medium confidence

* 1. STACK1

Đề bài

Graphical user interface, text, application

Description automatically generated with medium confidence

Code

#include <bits/stdc++.h>

using namespace std;

int main() {

string s;

stack<char> st;

cin >> s;

for (int i = 0 ; i < s.size() ; i++) {

if ((s[i] >= 'A' && s[i] <= 'Z') || (s[i] >= 'a' && s[i] <= 'z'))

st.push(s[i]);

if (s[i] == '\*') {

if (!st.empty()) {

cout << st.top();

st.pop();

}

}

}

return 0;

}

Testcase1

Text

Description automatically generated

Testcase2

Text

Description automatically generated with medium confidence

Testcase3

Text

Description automatically generated

* 1. NGOACDUNG

Đề bài

Text, letter, email

Description automatically generated

Code

#include <bits/stdc++.h>

using namespace std;

int main() {

stack<char> st;

char x;

while (cin >> x) {

if (x == '[' || x == '(' || x == '{')

st.push(x);

else {

if (!st.empty()) {

if (x == ']' && st.top() == '[')

st.pop();

else if (x == ')' && st.top() == '(')

st.pop();

else if (x == '}' && st.top() == '{')

st.pop();

}

else

st.push(x);

}

}

if (st.empty())

cout << "yes";

else

cout << "no";

return 0;

}

Testcase1

Text

Description automatically generated with medium confidence

Testcase2

A picture containing application

Description automatically generated

Testcase3

A picture containing text

Description automatically generated