//Problema 1

#include <iostream>

using namespace std;

class A {

protected:

int x;

public:

A(int i = 14) { x = i; }

};

class B : A {

public:

B()

: A(2)

{

}

B(B& b) { x = b.x - 14; }

void afisare() { cout << x; }

};

int main()

{

B b1, b2(b1);

b2.afisare();

return 0;

}

//problema 2

class A {

int x;

public:

A(int i) { x = i; }

int get\_x() { return x; }

int& set\_x(int i) { x = i; }

A operator=(A a1)

{

set\_x(a1.get\_x());

return a1;

}

};

class B : public A {

int y;

public:

B(int i)

: A(i)

{

y = i;

}

void afisare() { cout << y; }

};

int main()

{

B a(112), b, \*c;

cout << (b = a).get\_x();

(c = &a)->afisare();

return 0;

}

//problema 3

class A {

int x;

public:

A(int i = 2)

: x(i)

{

}

int get\_x() const { return x; }

};

class B : public A {

int\* y;

public:

B(int i = 2)

: A(i)

{

y = new int[i];

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

y[j] = 1;

}

B(B& b)

{

y = new int[b.get\_x()];

for (int i = 0; i < b.get\_x(); i++)

y[i] = b[i];

}

int& operator[](int i) const { return y[i]; }

};

ostream& operator<<(ostream& o, const B b)

{

for (int i = 0; i < b.get\_x(); i++)

o << b[i];

return o;

}

int main()

{

const B b(5);

cout << b;

return 0;

}

//problema 4

class D;

class B {

int x;

friend void f(B, D);

public:

B(int i = 0) { x = i; }

};

class D : public B {

public:

int y;

D(int i = 0, int j = 0)

: B(i)

{

y = j;

}

};

void f(B b, D d) { cout << b.x << " " << d.y; }

int main()

{

B b;

D d;

f(b, d);

}

//problema 5

class B

{ protected: int i;

public: B(int j=5) {cout << " cb "; i=j-2; }

~B(){ cout << " db ";}

int get\_i() { return i; } };

class D1: public B

{ int j;

public: D1(int k=10) {cout << " cd1 "; j=i-k+3; }

~D1(){ cout << " dd1 ";} };

class D2: public D1

{ int k;

public: D2(int l=15) {cout << " cd2 "; k=i-l+3; }

~D2(){ cout << " dd2 ";} };

D1 f(D1 d1, D2 d2) {return d1.get\_i()+d2.get\_i(); }

int main()

{ D2 ob2; D1 ob1(3);

cout<<f(ob1,ob2).get\_i()+ob2.get\_i();

return 0;

}

//problema 6

#include <iostream>

#include <typeinfo>

using namespace std;

class B

{ int i;

public: B() { i=1; }

int get\_i() { return i; }

};

class D: public B

{ int j;

public: D() { j=2; }

int get\_j() {return j; }

};

int main()

{ B \*p=new D;

cout<<p->get\_i();

if (typeid((B\*)p).name()=="B") cout<<((D\*)p)->get\_j();

return 0;

}

//problema 7

template <class T, class U>

T f(T x, U y)

{

return x + y;

}

int f(int x, int y)

{

return x - y;

}

int main()

{

int \*a = new int(3), b(23);

cout << \*f(a, b);

return 0;

}

//problema 8

class B {

int i;

public:

B() { i = 1; }

int get\_i() { return i; }

};

class D : B {

int j;

public:

D() { j = 2; }

int get\_j() { return j; }

};

int main()

{

B\* p = new D;

cout << p->get\_i();

if (typeid((B\*)p).name() == "D\*")

cout << ((D\*)p)->get\_j();

return 0;

}

//problema 9

class B

{ int i;

public: B() { i=80; }

virtual int get\_i() { return i; }

};

class D: public B

{ int j;

public: D() { j=27; }

int get\_j() {return j; }

};

int main()

{ D \*p=new B;

cout<<p->get\_i();

cout<<((D\*)p)->get\_j();

return 0;

}

//problema 10

class A

{

int x;

public:

A(int i = 25) { x = i; }

int& f() const { return x; }

};

int main()

{

A ob(5);

cout << ob.f();

return 0;

}

//problema 11

int f(int y)

{

if (y < 0)

throw y;

return y / 2;

}

int f(int y, int z)

{

if (y < z)

throw z - y;

return y / 2;

}

float f(int& y)

{

cout <<” y este referinta”;

return (float)y / 2;

}

int main()

{

int x;

try {

cout <<”Da - mi un numar par : ”;

cin >> x;

if (x % 2)

x = f(x, 0);

else

x = f(x);

cout <<”Numarul ”<< x <<” e bun !”<< endl;

}

catch (int i) {

cout <<”Numarul ”<< i <<” nu e bun !”<< endl;

}

return 0;

}

//problema 12

class cls1 {

int x;

public:

cls1 () {

x = 13;

}

int g() {

static int i; i++;

return (i+x);

}

};

class cls2 {

int x;

public:

cls2() {

x = 27;

}

cls1& f() {

cls1 ob; return ob;

}

};

int main() {

cls2 ob;

cout << ob.f().g();

return 0;

}

//problema 13

class A {

protected:

static int x;

public:

A(int i = 0) { x = i; }

virtual A schimb() { return (7 - x); }

};

class B : public A {

public:

B(int i = 0) { x = i; }

void afisare() { cout << x; }

};

int A::x = 5;

int main()

{

A\* p1 = new B(18);

\*p1 = p1->schimb();

((B\*)p1)->afisare();

return 0;

}

//problema 14  
class B {

public:

int x;

B(int i = 16) { x = i; }

B f(B ob) { return x + ob.x; }

};

class D : public B {

public:

D(int i = 25) { x = i; }

D f(D ob) { return x + ob.x + 1; }

void afisare() { cout << x; }

};

int main()

{

D \*p1 = new D, \*p2 = new B, \*p3 = new D(p1->f(\*p2));

cout << p3->x;

return 0;

}

//problema 15

class B {

protected:

int x;

public:

B(int i = 28) { x = i; }

virtual B f(B ob) { return x + ob.x + 1; }

void afisare() { cout << x; }

};

class D : public B {

public:

D(int i = -32)

: B(i)

{

}

B f(B ob) { return x + ob.x - 1; }

};

int main()

{

B \*p1 = new D, \*p2 = new B, \*p3 = new B(p1->f(\*p2));

p3->afisare();

return 0;

}

//problema 16

class cls1 {

public:

int x;

cls1(int i = 20) { x = i; }

};

class cls2 {

public:

int y;

cls2(int i = 30) { y = i; }

operator cls1()

{

cls1 ob;

ob.x = y;

return ob;

}

};

cls1 f(cls1 ob)

{

ob.x++;

return ob;

}

int main()

{

cls1 ob1;

f(ob1);

cout << ob1.x;

cls2 ob2;

f(ob2);

cout << ob2.y;

return 0;

}

//problema 17

class B {

protected:

static int x;

int i;

public:

B()

{

x++;

i = 1;

}

~B() { x--; }

static int get\_x() { return x; }

int get\_i() { return i; }

};

int B::x;

class D : public B {

public:

D()

{

x++;

i++;

}

~D()

{

x--;

i--;

}

int f1(B o) { return 5 + get\_i(); }

};

int f(B\* q)

{

return (q->get\_x()) + 1;

}

int main()

{

B\* p = new B[10];

cout << f(p);

delete[] p;

p = new D;

cout << p->f1(p);

delete p;

cout << D::get\_x();

return 0;

}

//problema 18

class B {

protected:

static int x;

int i;

public:

B()

{

x++;

i = 1;

}

~B() { x--; }

static int get\_x() { return x; }

int get\_i() { return i; }

};

int B::x;

class D : public B {

public:

D()

{

x++;

i++;

}

~D()

{

x--;

i--;

}

int f1(B o) { return 5 + get\_i(); }

};

int f(B\* q)

{

return (q->get\_x()) + 1;

}

int main()

{

B\* p = new B[10];

cout << f(p);

delete[] p;

p = new D;

cout << p->f1(p);

delete p;

cout << D::get\_x();

return 0;

}

//problema 19

class cls {

int x;

public:

cls(int i) { x = i; }

int set\_x(int i)

{

int y = x;

x = i;

return y;

}

int get\_x() { return x; }

};

int main()

{

cls\* p = new cls[10];

int i = 0;

for (; i < 10; i++)

p[i].set\_x(i);

for (i = 0; i < 10; i++)

cout << p[i].get\_x();

return 0;

}

//problema 20

class cls {

public:

int x, y;

cls(int i = 0, int j = 0)

{

x = i;

y = j;

}

};

void main()

{

cls a, b, c[3] = { cls(1, 1), cls(2, 2), a };

cout << c[2].x;

}

//problema 21

class cls {

public:

int x;

cls() { x = 3; }

void f(cls& c) { cout << c.x; }

};

void main()

{

const cls d;

f(d);

}

//problema 22

class problema {

int i;

public:

problema(int j = 5) { i = j; }

void schimba() { i++; }

void afiseaza() { cout << "starea curenta " << i << "\n"; }

};

problema mister1() { return problema(6); }

void mister2(problema& o)

{

o.afiseaza();

o.schimba();

o.afiseaza();

}

int main()

{

mister2(mister1());

return 0;

}

//problema 23

class B {

int x;

public:

B(int i = 2)

: x(i)

{

}

int get\_x() const { return x; }

};

class D : public B {

int\* y;

public:

D(int i = 2)

: B(i)

{

y = new int[i];

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

y[j] = 1;

}

D(D& a)

{

y = new int[a.get\_x()];

for (int i = 0; i < a.get\_x(); i++)

y[i] = a[i];

}

int& operator[](int i) { return y[i]; }

};

ostream& operator<<(ostream& o, const D& a)

{

for (int i = 0; i < a.get\_x(); i++)

o << a[i];

return o;

}

int main()

{

D ob(5);

cout << ob;

return 0;

}

//problema 24

class cls {

int x;

public:

cls(int i = 25) { x = i; }

int f();

};

int cls::f() { return x; }

int main()

{

const cls d(15);

cout << d.f();

return 0;

}

//problema 25

class A

{ int x;

static int y;

public: A(int i,int j):x(i),y(j){}

int f() const;};

int A::y;

int A::f() const {return y;}

int main()

{ const A a(21,2);

cout<<a.f();

return 0;

}

//problema 26

template<class X>

int functie(X x, X y)

{ return x+y;

}

int functie(int & x, int \*y)

{ return x-\*y;

}

int main()

{ int a=7, \*b=new int(4);

cout<<functie(a,b);

return 0;

}

//problema 27

class A

{

int i;

protected: static int x;

public: A(int j=7) {i=j;x=j;}

int get\_x() {return x;}

int set\_x(int j) {int y=x; x=j; return y;}

A operator=(A a1) {set\_x(a1.get\_x()); return a1;}

};

int A::x=15;

int main()

{

A a(212),b;

cout<<(b=a).get\_x();

return 0;

}

//problema 28

class A

{

int x;

public: A(int i = 17) { x = i; }

int get\_x() { return x; }

};

class B

{

int x;

public: B(int i = -16) { x = i; }

operator A() { return x; }

int get\_x() { return x; }

};

int main()

{

B a;

A b = a;

cout << b.get\_x();

return 0;

}

//problema 29

class cls {

public:

float sa;

cls(float s = 0) { sa = s; }

operator float() { return sa; }

float f(float c) { return (sa \* (1 + c / 100)); }

};

int main()

{

cls p(100);

cout << p.f(p);

}

//problema 30

class B

{ protected: static int x;

int i;

public: B() { x++; i=1; }

~B() { x--; cout << "b";}

int get\_x() { return x; }

int get\_i() { return i; } };

int B::x;

class D: public B

{ public: D() { x++; }

~D() { x--; cout << "d";} };

int f(B \*q)

{ return (q->get\_x())+1; }

int main()

{ B \*p=new B[10];

cout<<f(p);

delete[] p;

p=new D;

cout<<f(p);

delete p;

cout<<D::get\_x();

return 0;

}

//problema 31

template <class T, class U>

T fun(T x, U y)

{

return x + y;

}

int fun(int x, int y)

{

return x - y;

}

int fun(int x)

{

return x + 1;

}

int main()

{

int \*a = new int(10), b(5);

cout << fun(a, b);

return 0;

}

//problema 32

class cls {

int vi;

public:

cls(int v = 18) { vi = v; }

operator int() { return vi; }

cls operator++()

{

vi++;

return \*this;

}

cls operator++(int);

};

cls cls::operator++(int)

{

cls aux = \*this;

vi++;

return aux;

}

int main()

{

cls p(20);

int x = p++, y = ++p;

cout << "x=" << x << ", y=" << y << endl;

}

//problema 33

class A

{

protected: int x;

public: A(int i = -16) { x = i; }

virtual A f(A a) { return x + a.x; }

void afisare() { cout << x; }

};

class B: public A

{

public: B(int i=3):A(i){}

A f(A a) { return x + a.x + 1; }

};

int main()

{

A \*p1 = new B, \*p2 = new A, \*p3 = new A(p1->f(\*p2));

p3->afisare();

return 0;

}

//problema 34

class A {

static int x;

public:

A(int i = 0) { x = i; }

int get\_x() { return x; }

int& set\_x(int i) { x = i; }

A operator=(A a1)

{

set\_x(a1.get\_x());

return a1;

}

};

int main()

{

A a(212), b;

cout << (b = a).get\_x();

return 0;

}

//problema 35

class B {

int i;

public:

B() { i = 1; }

int get\_i() { return i; }

};

class D : public B {

int j;

public:

D() { j = 2; }

int get\_j() { return j; }

};

int main()

{

B\* p;

int x = 0;

if (x)

p = new B;

else

p = new D;

if (typeid(p).name() == "D\*")

cout << ((D\*)p)->get\_j();

return 0;

}

//problema 36

class vector {

int \*p, nr;

public:

operator int() { return nr; }

vector(int);

};

vector::vector(int n)

{

p = new int[n];

nr = n;

while (n--)

p[n] = n;

}

void f(int i)

{

while (i--)

cout << i << endl;

}

int main()

{

vector x(10);

f(x);

}

//problema 37

template<class X>void test(X &a, X &b)

{

X temp;

temp=a;

a=b;

b=temp;

cout<<"ttest\n";

}

void test(int &c,int &d)

{

int temp;

temp=c;

c=d;

d=temp;

cout<<"test 1\n";

}

void test(int e,int f)

{

int temp;

temp=e;

e=f;

f=temp;

cout<<"test 2\n";

}

void test(int g,int \*h)

{

int temp;

temp=g;

g=\*h;

\*h=temp;

cout<<"test 3\n";

}

int main()

{

int a=5,b=15,c=25,\*d=&a;

test(a,b);

test(c,d);

return 0;

}

//problema 38

class A

{ protected: static int x;

public: A(int i=1) {x=i; }

int get\_x() { return x; }

int& set\_x(int i) { int y=x; x=i; return y;}

A operator=(A a1) { set\_x(a1.get\_x()); return a1;}

} a(33);

int main()

{ A a(18), b(7);

cout<<(b=a).set\_x(27);

return 0;

}

//problema 39

template <class T>

int f(T x, T y)

{

return x + y;

}

int f(int x, int y)

{

return x - y;

}

int main()

{

int a = 5;

float b = 8.6;

cout << f(a, b);

return 0;

}

//problema 40

class A

{

int i;

public: A() {i=1;}

virtual int get\_i() {return i;}

};

class B: public A

{

int j;

public: B() {j=2;}

int get\_i() { return A::get\_i()+j;}

};

int main()

{

const int i=cin.get();

if(i%3) {A o;}

else {B o;}

cout<<o.get\_i();

return 0;

}

//problema 41

class cls {

static int x;

public:

cls(int i = 1) { x = i; }

cls f(cls a) { return x + a.x; }

static int g() { return f() / 2; }

};

int cls::x = 7;

int main()

{

cls ob;

cout << cls::g();

return 0;

}

//problema 42

class cls

{ int x;

public: cls(int i=2) { x=i; }

int set\_x(int i) { int y=x; x=i; return y; }

int get\_x(){ return x; } };

int main()

{ cls \*p=new cls[15];

for(int i=2;i<8;i++) p[i].set\_x(i);

for(int i=1;i<6;i++) cout<<p[i].get\_x();

return 0;

}

//problema 43

class A

{

protected: int x;

public: A(int i = -16) { x = i; }

virtual A f(A a) { return x + a.x; }

void afisare() { cout << x; }

};

class B: public A

{

public: B(int i=3):A(i){}

A f(A a) { return x + 1; }

B operator+ (B a) { return x + a.x; }

};

int main()

{

B a; int b = -21;

a += b;

cout << b;

return 0;

}

//problema 44

struct cls {

int x;

public:

int set\_x(int i)

{

int y = x;

x = i;

return x;

}

int get\_x() { return x; }

};

int main()

{

cls\* p = new cls[100];

int i = 0;

for (; i < 50; i++)

p[i].set\_x(i);

for (i = 5; i < 20; i++)

cout << p[i].get\_x() << " ";

return 0;

}

//problema 45

class B {

int x;

public:

B(int v) { v = x; }

int get\_x() { return x; }

};

class D : private B {

int y;

public:

D(int v)

: B(v)

{

}

int get\_x() { return x; }

};

int main()

{

D d(10);

cout << d.get\_x();

}

//problema 46

class X { int i;

public: X(int ii = 5) { i = ii; cout<< i<< " ";};

const int tipareste(int j) const { cout<<i<< " "; return i+j; }; };

int main()

{ X O (7);

O.tipareste(5);

X &O2=O;

O2.tipareste(6);

const X\* p=&O;

cout<<p->tipareste(7);

return 0;

}

//problema 47

class B {

int i;

public:

B() { i = 1; }

virtual int get\_i() { return i; }

};

class D : virtual public B {

int j;

public:

D() { j = 2; }

int get\_i() { return B::get\_i() + j; }

};

class D2 : virtual public B {

int j2;

public:

D2() { j2 = 3; }

int get\_i() { return B::get\_i() + j2; }

};

class MM : public D, public D2 {

int x;

public:

MM() { x = D::get\_i() + D2::get\_i(); }

int get\_i() { return x; }

};

int main()

{

B\* o = new MM();

cout << o->get\_i() << "\n";

MM\* p = dynamic\_cast<MM\*>(o);

if (p)

cout << p->get\_i() << "\n";

D\* p2 = dynamic\_cast<D\*>(o);

if (p2)

cout << p2->get\_i() << "\n";

return 0;

}

//problema 48

class B {

protected:

static int x;

int i;

public:

B()

{

x++;

i = 1;

}

~B() { x--; }

static int get\_x() { return x; }

int get\_i() { return i; }

};

int B::x;

class D : public B {

public:

D() { x++; }

~D() { x--; }

};

int f(B\* q)

{

return (q->get\_x()) + 1;

}

int main()

{

B\* p = new B[10];

cout << f(p);

delete[] p;

p = new D;

cout << f(p);

delete p;

cout << D::get\_x();

return 0;

}

//problema 49

class A

{ int i;

public: A(int x=2):i(x+1) {}

virtual int get\_i() { return i; }};

class B: public A

{ int j;

public: B(int x=20):j(x-2) {}

virtual int get\_j() {return A::get\_i()+j; }};

int main()

{ A o1(5);

B o2;

cout<<o1.get\_i()<<" ";

cout<<o2.get\_j()<<" ";

cout<<o2.get\_i();

return 0;

}

//problema 50

class B {

int x;

public:

B(int i = 7) { x = i; }

int get\_x() { return x; }

operator int() { return x; }

};

class D : public B {

public:

D(int i = -12)

: B(i)

{

}

D operator+(D a) { return get\_x() + a.get\_x() + 1; }

};

int main()

{

D a;

int b = 18;

b += a;

cout << b;

return 0;

}

//problema 51

class cls {

int vi;

public:

cls(int v = 37) { vi = v; }

friend int& f(cls);

};

int& f(cls c) { return c.vi; }

int main()

{

const cls d(15);

f(d) = 8;

cout << f(d);

return 0;

}

//problema 52

class B {

public:

int x;

B(int i = 16) { x = i; }

B f(B ob) { return x + ob.x; }

};

class D : public B {

public:

D(int i = 25) { x = i; }

B f(B ob) { return x + ob.x + 1; }

void afisare() { cout << x; }

};

int main()

{

B \*p1 = new D, \*p2 = new B, \*p3 = new B(p1->f(\*p2));

cout << p3->x;

return 0;

}

//problema 53

class B {

int i;

public:

B(int x) { i = x + 1; }

int get\_i() { return i; }

};

class D : public B {

int j;

public:

D()

: B(1)

{

j = i + 2;

}

int get\_j() { return j; }

};

int main()

{

B\* p = new D[10];

cout << p->get\_i();

if (typeid((B\*)p).name() == "D\*")

cout << ((D\*)p)->get\_j();

return 0;

}

//problema 54

class Y;

class Z;

class X {

int x;

public:

X(int n = 0) { x = n; }

friend Y;

};

class Y {

int y;

friend Z;

};

class Z {

public:

void f(X u) { cout << u.x; }

};

void main()

{

X a;

Z b;

b.f(a);

}

//prolema 55

template <class tip>

tip dif(tip x, tip y)

{

return x - y;

}

unsigned dif(unsigned x, unsigned y)

{

return x >= y ? x - y : y - x;

}

int main()

{

unsigned i = 7, j = 8;

cout << dif(i, j);

return 0;

}

//problema 56

class A {

public:

int x;

A(int i = -13) { x = i; }

};

class B : virtual public A {

public:

B(int i = -15) { x = i; }

};

class C : virtual public A {

public:

C(int i = -17) { x = i; }

};

class D : virtual public A {

public:

D(int i = -29) { x = i; }

};

class E : public B, public D, public C {

public:

int y;

E(int i, int j)

: D(i)

, B(j)

{

y = x + i + j;

}

E(E& ob) { y = ob.x - ob.y; }

};

int main()

{

E e1(5, 10), e2 = e1;

cout << e2.y;

return 0;

}

//problema 57

class cls

{ int x;

public: cls(int y) {x=y; }

friend int operator\*(cls a,cls b){return (a.x\*b.x); } };

int main()

{ cls m(100),n(15);

cout << m\*n;

return 0; }

//problema 58

class A {

int x;

public:

A(int i = 2)

: x(i)

{

}

int get\_x() const { return x; }

};

class B : public A {

int\* y;

public:

B(int i = 2)

: A(i)

{

y = new int[i];

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

y[j] = 1;

}

int& operator[](int i) const { return y[i]; }

};

ostream& operator<<(ostream& o, const B &b)

{

for (int i = 0; i < b.get\_x(); i++)

o << b[i];

return o;

}

int main()

{

const B b(5);

b[2] = 3; b[4] = 5;

cout << b;

return 0;

}

//problema 59

class B {

protected:

int x;

public:

B(int i = 12) { x = i; }

virtual B f(B ob) { return x + ob.x + 1; }

void afisare() { cout << x; }

};

class D : public B {

public:

D(int i = -15)

: B(i - 1)

{

x++;

}

B f(B ob) { return x - 2; }

};

int main()

{

B \*p1 = new D, \*p2 = new B, \*p3 = new B(p1->f(\*p2));

p3->afisare();

return 0;

}

//problema 60

class cls1 {

protected:

int x;

public:

cls1(int i = 10) { x = i; }

int get\_x() { return x; }

};

class cls2 : cls1 {

public:

cls2(int i)

: cls1(i)

{

}

};

int main()

{

Cls2 d(37);

cout << d.get\_x();

return 0;

}

//problema 61

class A

{ protected: static int x;

private: int y;

public: A(int i) { x=i; y=-i+4; }

int put\_x(A a) { return a.x+a.y; } };

int A::x=7;

int main()

{ A a(10);

cout<<a.put\_x(20);

return 0;

}

//problema 62

class A

{ protected: int x;

public: A(int i):x(i){}

int get\_x(){ return x; } };

class B: A

{ protected: int y;

public: B(int i,int j):y(i),A(j){}

int get\_y(){ return get\_x()+y; } };

class C: protected B

{ protected: int z;

public: C(int i,int j,int k):z(i),B(j,k){}

int get\_z(){ return get\_x()+get\_y()+z; } };

int main()

{ C c(5,6,7);

cout<<c.get\_z();

return 0;

}

//problema 63

class cls

{

int x;

public: cls(int i=3) {x=i;}

int &f() const{ return x;}

};

int main()

{

const cls a(-3);

int b=a.f();

cout<<b;

return 0;

}

//problema 64

class cls {

int \*v, nr;

public:

cls(int i)

{

nr = i;

v = new int[i];

for (int j = 1; j < nr; j++)

v[j] = 0;

}

int size() { return nr; }

int& operator[](int i) { return \*(v + i); }

};

int main()

{

cls x(10);

x[4] = -15;

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

cout << x[i];

return 0;

}

//problema 65

class A

{ static int \*x;

public: A() {}

int get\_x() { return (++(\*x))++; } };

int \*A::x(new int(19));

int main()

{ A \*p=new A,b;

cout<<b.get\_x()<<" ";

cout<<p->get\_x();

return 0;

}

//problema 66

class B {

int x;

public:

B(int i = 10) { x = i; }

int get\_x() { return x; }

};

class D : public B {

public:

D(int i)

: B(i)

{

}

D operator+(const D& a) { return x + a.x; }

};

int main()

{

D ob1(7), ob2(-12);

cout << (ob1 + ob2).get\_x();

return 0;

}

//problema 67

class cls1 {

protected:

int x;

public:

cls1() { x = 13; }

};

class cls2 : public cls1 {

int y;

public:

cls2() { y = 15; }

int f(cls2 ob) { return (ob.x + ob.y); }

};

int main()

{

cls2 ob;

cout << ob.f(ob);

return 0;

}

//problema 68

class cls1

{ protected: int x;

public: cls1(int i=10) { x=i; }

int get\_x() { return x;} };

class cls2: cls1

{ public: cls2(int i):cls1(i) {} }; int main()

{ cls2 d(37);

cout<<d.get\_x();

return 0; }

//problema 69

template<class X, class Y>

X f(X x, Y y)

{

return x+y;

}

int \*f(int \*x,int y)

{

return x-y;

}

int main()

{

int \*a=new int(200), \*b=a;

cout<<\*f(a,b);

return 0;

}

//problema 70

class A

{

int valoare;

public: A(int param1=3):valoare(param1){}

int getValoare(){return this->valoare;}

};

int main()

{

A vector[]={\*(new A(3)),\*(new A(4)),\*(new A(5)),\*(new A(6)) };

cout<<vector[2].getValoare();

return 0;

}

//problema 71

class A

{

int i;

public: A() { i = 1; }

int get\_i() { return i; }

};

class B: public A

{

int j;

public: B() { j = 2; }

int get\_j() { return j; }

};

int main()

{

A \*p;

int x = 0;

if (x) p = new A;

else p = new B;

if (typeid(p).name() == typeid(B\*).name()) cout << ((B\*)p)->get\_j();

else cout << "tipuri diferite";

return 0;

}

//problema 72

struct X { int i;

public: X(int ii ) { i = ii; };

int f1() { return i; }

X f2() const { int i=this->f1(); return X(34-i); }};

const X f3() { return X(16); }

int f4(const X& x) { return x.f1(); }

int main() {X ob(11);

cout<<f4(ob.f2().f1());

return 0;

}