

„Mit ír ki?” feladatok megoldásokkal

A feladatgyűjtemény 2006 és 2015 között ZH feladatként kiadott C++ feladatokat és megoldásukat tartalmazza. A megoldásokat sárga kiemeléssel jelöltük.

Tartalomjegyzék

1. Feladat	3
2. Feladat	3
3. Feladat	4
4. Feladat	4
5. Feladat	5
6. Feladat	5
7. Feladat	6
8. Feladat	6
9. Feladat	7
10. Feladat	7
11. Feladat	8
12. Feladat	8
13. Feladat	10
14. Feladat	11
15. Feladat	11
16. Feladat	12
17. Feladat	13
18. Feladat	13
19. Feladat	14
20. Feladat	15
21. Feladat	15
22. Feladat	15
23. Feladat	16
24. Feladat	16
25. Feladat	17
26. Feladat	17
27. Feladat	18
28. Feladat	19
29. Feladat	21
30. Feladat	22
31. Feladat	22

32. Feladat	23
33. Feladat	23
34. Feladat	23
35. Feladat	24
36. Feladat	24
37. Feladat	25
38. Feladat	26
39. Feladat	27
40. Feladat	28
41. Feladat	29
42. Feladat	29
43. Feladat	30
44. Feladat	32
45. Feladat	33
46. Feladat	34
47. Feladat	34
48. Feladat	34
49. Feladat	35
50. Feladat	35
51. Feladat	36
52. Feladat	36
53. Feladat	36

1. Feladat

Mit ír ki a szabványos kimenetre a program? Válaszához használja a négyzetrácsos területet!

```
#include <iostream>
using namespace std;
```

```
class A {
public:
    A(const int i = 0)      { cout << 'k' << i; }
    A(const A& a)          { cout << 'c'; }
    virtual ~A()           { cout << 'd'; }
};

class B :public A {
    const char *pa;
public:
    B(const char *n = "") :pa(n), A(-1) { cout << 'K' << pa; }
    B(int i) :A(i)             { cout << 'L'; }
    ~B()                       { cout << 'D'; }
};

int main() {
    B b("234");                cout << '\n';
    A a = b;                   cout << '\n';
    A* p2 = new B(2);          cout << '\n';
    delete p2;                 cout << '\n';

    return(0);
}
```

k	-	1	K	2	3	4				
c										
k	2	L								
D	d									
d	D	d								

2. Feladat

Mit ír ki a szabványos kimenetre a program? Válaszához használja a négyzetrácsos területet!

```
#include <iostream>
using namespace std;
```

```
class A {
public:
    A(const int i = 0)      { cout << 'k' << i; }
    A(const A& a)          { cout << 'c'; }
    void operator++()      { cout << 'p'; }
    virtual ~A()           { cout << 'd'; }
};

class B :public A {
public:
    B(const char *n = "2---") { cout << 'K' << n; }
    B(int i) :A(i)             { cout << 'K'; }
    B(const B& b) :A(b)        { cout << 'C'; }
    void operator++(int i2) { cout << 'P'; }
    ~B()                       { cout << 'D'; }
};

int main() {
    B b1("2---");              cout << '\n';
    B b2 = b1;                 cout << '\n';
    A* p1 = new B(2);          cout << '\n';
    ++(*p1);                   cout << '\n';
    delete p1;                 cout << '\n';
    return(0);
}
```

k	0	K	-	-	-					
c	C									
k	2	K								
P										
D	d									
D	d	D	d							

3. Feladat

Mit ír ki a szabványos kimenetre a program? Válaszához használja a négyzettrácsos területet!

```
#include <iostream>
using namespace std;
class A {
public:
    A(const int i = -1)          { cout << 'k' << i; }
    A(const A& a)                { cout << 'c'; }
    virtual void operator++(int) { cout << 'p'; }
    ~A()                         { cout << 'd'; }
};
class B :public A {
public:
    B(const char *n = "2j72")    { cout << 'K' << n; }
    B(int i) :A(i)                { cout << 'K'; }
    B(const B& b) :A(b)           { cout << 'C'; }
    void operator++(int)         { cout << 'P'; }
    ~B()                         { cout << 'D'; }
};
int main() {
    B b1("2j7");                 cout << '\n';
    B b2 = b1;                   cout << '\n';
    A* p1 = new B(2);            cout << '\n';
    (*p1)++;                     cout << '\n';
    delete p1;                   cout << '\n';
    return(0);
}
```

k	-	1	K	2	j	7														
c	C																			
k	2	K																		
P																				
d																				
D	d	D	d																	

4. Feladat

Mit ír ki a szabványos kimenetre a program? Válaszához használja a négyzettrácsos területet!

```
#include <iostream>
using namespace std;
class A {
public:
    A(const int i = 0)          { cout << 'k' << i; }
    A(const A& a)                { cout << "c-q"; }
    virtual void f(A& a)         { cout << 'f'; }
    ~A()                         { cout << 'd'; }
};
class B :public A {
    const char *pr;
public:
    B(const char *n = "q") :pr(n), A(65) { cout << 'K' << pr; }
    B(const B& b)                { cout << 'C'; }
    void f(A& a)                 { cout << 'F'; }
    ~B()                         { cout << 'D'; }
};
int main() {
    B b("lar");                  cout << '\n';
    A a = b;                     cout << '\n';
    A* p65 = new B;              cout << '\n';
    (*p65).f(a);                 cout << '\n';
    delete p65;                  cout << '\n';
    return(0);
}
```

k	6	5	K	1	a	r															
c	-	q																			
k	6	5	K	q																	
F																					
d																					
d	D	d																			

5. Feladat

Mit ír ki a szabványos kimenetre az alábbi program? Válaszához használja a négyzetrácsos területet!

```
#include <iostream>
using namespace std;
class A {
    const char *s;
public:
    A(const char *s = "D") :s(s)    { cout << s << 'k'; f(); }
    virtual ~A()                  { cout << 'd'; }
    virtual void f()              { cout << "f"; }
};
class B : public A {
public:
    B(const char *s = "C++")      { cout << 'K'; f(); }
    B(const B& b)                 { cout << 'C'; f(); }
    B(const A)                    { cout << 'M'; }
    ~B()                          { cout << 'D'; }
    void f()                      { cout << "R"; }
};
int main() {
    A me6("QBX734"); cout << endl;
    B b1("6");        cout << endl;
    B b2 = b1;       cout << endl;
    B b3(me6);       cout << endl;
    delete new A;    cout << endl;
    return(0);
}
```

Q	B	X	7	3	4	k	f							
D	k	f	K	R										
D	k	f	C	R										
D	k	f	M	d										
D	k	f	d											
D	d	D	d	D	d	d								

6. Feladat

Mit ír ki a szabványos kimenetre az alábbi program? Válaszához használja a négyzetrácsos területet!

```
#include <iostream>
using namespace std;
class A {
    const char *s;
public:
    A(const char *s = "A") :s(s)    { cout << s << 'k'; }
    virtual ~A()                  { cout << 'd'; }
};
class B :public A {
public:
    B(const char *s = "ABCDEF") :A(s){ cout << 'K'; }
    B(const B&)                  { cout << 'C'; }
    ~B()                          { cout << 'D'; }
};
int main() {
    A ab4;                        cout << endl;
    B b1("lo4");                  cout << endl;
    B b2 = b1;                    cout << endl;
    A *ap = new B("lo4"); cout << endl;
    delete ap;                    cout << endl;
    return(0);
}
```

A	k													
l	o	4	k	K										
A	k	C												
l	o	4	k	K										
D	d													
D	d	D	d	d										

Ha felüldefiniáljuk a másoló konstruktort és abban explicit nem hívjuk meg az őszülő másoló konstruktort, akkor az őszülő alapértelmezett konstruktora hívódik meg!

7. Feladat

Mit ír ki a szabványos kimenetre az alábbi program? Válaszához használja a négyzetrácsos területet!

```
#include <iostream>
using namespace std;
class A {
    const char *s;
public:
    A(const char *s = "A") :s(s)    { cout << s << 'k'; }
    virtual ~A()                  { cout << 'd'; }
};
class B {
    A a;
public:
    B(const char *s = "C++") :a(s)  { cout << 'K'; }
    B(const B& b)                 { cout << 'C'; }
    B(const A)                    { cout << 'M'; }
    ~B()                          { cout << 'D'; }
};
int main() {
    A ab4("ABCDEF"); cout << endl;
    B b1("C++4");    cout << endl;
    B b2 = b1;      cout << endl;
    B b3(ab4);     cout << endl;
    delete new A;  cout << endl;
    return(0);
}
```

A	B	C	D	E	F	k								
C	+	+	4	k	K									
A	k	C												
A	k	M	d											
A	k	d												
D	d	D	d	D	d	d								

8. Feladat

Mit ír ki a szabványos kimenetre az alábbi program? Válaszához használja a négyzetrácsos területet!

```
#include <iostream>
using namespace std;
class Alap {
    int alap;
public:
    Alap(int i) :alap(i)    { cout << alap << 'k'; }
    Alap(const Alap&) :alap(100) { cout << alap << 'c'; }
    virtual ~Alap()        { cout << 'd'; }
};
class Uj :public Alap {
    const char *s;
public:
    Uj(const char *s = "B", int al = 0)
        :s(s), Alap(al)    { cout << s << "Uk"; }
    virtual ~Uj()          { cout << "Ud"; }
};
int main() {
    Uj e45("E4ESB1"); cout << endl;
    Uj b1 = e45;      cout << endl;
    Alap *uj = new Uj;cout << endl;
    delete uj;       cout << endl;
    uj = new Alap(5); cout << endl;
    return(0);
}
```

0	k	E	4	E	S	B	1	U	k					
1	0	0	c											
0	k	B	U	k										
U	d	d												
5	k													
U	d	d	U	d	d									

9. Feladat

Mit ír ki a szabványos kimenetre az alábbi program? Válaszához használja a négyzetrácsos területet!

```
#include <iostream>
using namespace std;

class Alap {
    int alap;
public:
    Alap(int i = 0) :alap(i)          { cout << alap << 'k';}
    Alap(const Alap&) :alap(360)     { cout << alap << 'c';}
    Alap& operator=(const Alap& a)   { cout << a.alap << 'e'; return
*this;}
    virtual ~Alap()                  { cout << 'd'; }
};

class Uj :public Alap {
    const char *s;
public:
    Uj(const char *s = "A", int al = 0)
        :s(s), Alap(al)              { cout << s << "Uk"; }
    Uj(const Uj& u) : s(u.s)          { cout << s << "Uc"; }
    ~Uj()                             { cout << "Ud"; }
};

int main() {
    Uj me0("MEGOLD", 0);             cout << endl;
    Uj b1 = me0;                     cout << endl;
    b1 = me0;                        cout << endl;
    Alap* p = new Alap[2];           cout << endl;
    delete[] p;                      cout << endl;
    return(0);
}
```

0	k	M	E	G	O	L	D	U	k		
0	k	M	E	G	O	L	D	U	c		
0	e										
0	k	0	k								
d	d										
U	d	d	U	d	d						

10. Feladat

Mit ír ki a szabványos kimenetre az alábbi program? Válaszához használja a négyzetrácsos területet!

```
#include <iostream>
using namespace std;

class Adat {
    int adat;
public:
    Adat(int i = 0) :adat(i)          { cout << adat << 'k';}
    Adat(const Adat&) :adat(4)       { cout << adat << 'c';}
    Adat& operator=(const Adat& a)   { cout << a.adat << 'e'; return
*this;}
    ~Adat()                           { cout << 'd'; }
};

class Ma {
    Adat a;
    const char *s;
public:
    Ma(const char *s = "C", int al = 0)
        :s(s), a(al)                 { cout << s << "Mk"; }
    ~Ma()                             { cout << "Md"; }
};
```

```
};

int main() {
    Ma me6("MEGOLD", 6);    cout << endl;
    Ma b1 = me6;            cout << endl;
    b1 = me6;              cout << endl;
    Ma* p = new Ma[1];     cout << endl;
    delete[] p;           cout << endl;
    return(0);
}
```

6	k	M	E	G	O	L	D	M	k					
4	c													
6	e													
0	k	C	M	k										
M	d	d												
M	d	d	M	d	d									

11. Feladat

Mit ír ki a szabványos kimenetre az alábbi program? Válaszához használja a négyzetrácsos területet!

```
#include <iostream>
using namespace std;

class Adat {
    int adat;
public:
    Adat(int i = 9) :adat(i)      { cout << adat << 'k';}
    Adat(const Adat&) :adat(2)   { cout << adat << 'c';}
    Adat& operator=(const Adat& a) { cout << a.adat << 'e';return *this;}
    ~Adat()                       { cout << 'd'; }
};

class Ma {
    Adat *ap;
    const char *s;
    Ma(const Ma&);
    Ma& operator=(const Ma&);
public:
    Ma(const char *s = "A", int i = 0)
        :s(s)                       { ap = new Adat(i); cout << s <<"Mk";}
    ~Ma()                             { delete ap; cout << "Md"; }
};

int main() {
    Ma me6("MEGOLD", 6);    cout << endl;
    Adat a1, a2(6);        cout << endl;
    a1 = a2;              cout << endl;
    Ma* p = new Ma[1];     cout << endl;
    delete[] p;           cout << endl;
    return(0);
}
```

6	k	M	E	G	O	L	D	M	k					
9	k	6	k											
6	e													
0	k	A	M	k										
d	M	d												
d	d	d	M	d										

12. Feladat

Mit ír ki a szabványos kimenetre az alábbi program? Válaszához használja a négyzetrácsos területet!

```
#include <iostream>
using namespace std;
class Alap {
    int alap;
public:
    Alap(int i) :alap(i)          { cout << alap << 'k';}
    Alap(const Alap& a) :alap(a.alap) { cout << alap << 'c';}
    operator bool() { return alap == 0; }
    ~Alap()                       { cout << 'd'; }
};
```



```
class Masik {
    Alap a;
    const char *m;
public:
    Masik(int a, const char *m) :a(a), m(m) { cout << m << 'M'; }
    void operator=(const Masik& e) { a = e.a; cout << '=' << e.m; }
    ~Masik() { cout << 'D'; }
};
```

```
int main() {
    Masik m1(1, "MEG123");      cout << endl;
    Masik m2 = m1;              cout << endl;
    m1 = m2;                    cout << endl;
    Alap a1(0);                 cout << endl;
    if (a1)                     cout << "g12" << endl;
    else                         cout << "me" << endl;
    return(0);
}
```

1	k	M	E	G	1	2	3	M	
1	c								
=	M	E	G	1	2	3			
0	k								
g	1	2							
*									
d	D	d	D	d					

* Az utolsó sort itt is elfogadtuk, mivel az if egyik ágán az endl nem íródik ki.

13. Feladat

Mit ír ki a szabványos kimenetre az alábbi program? Válaszához használja a négyzetrácsos területet! A soremelések segítségül szerepelnek a programban, de nem szükséges azokkal foglalkoznia. Abba a sorba írjon, amelyik sorban a kiírás keletkezik, ez lehet egy zárójelet tartalmazó sor is! A többit hagyja üresen!

```
#include <iostream>
#include <string>
using namespace std;

class Hiba {
    int h;
public:
    Hiba(int h) :h(h) { cout << 'H' << endl << endl;}
    int hibaKod() { return h; };
};

struct A0 {
    A0() { cout << '1'; }
    ~A0() { cout << 'd'; }
};

class Alap : public A0 {
    int al;
    string n;
public:
    Alap(int i, const char *p = "N12300") : n(p) {
        if (i == 0) throw Hiba(1);
        al = i;
        cout << i << 'k';
    }
    Alap(const Alap& a) :al(a.al) { cout << al << 'C' << endl << endl;}
    Alap& operator=(const Alap& a) { al = a.al; cout << a.n; return
*this; }
    ~Alap() { cout << 'D'; }
};
```

```

int main() {
    Alap a0(52);          cout << endl;
    try {                cout << endl;
        Alap a1(1, "23o"); cout << endl;
        a0 = a1;         cout << endl;
        Alap a2(0,"n1"); cout << endl;
        Alap a3 = a1;    cout << endl;
    } catch (Hiba& h) {  cout << endl;
        cout << h.hibaKod(); cout << endl;
    }                    cout << endl;
    return(0);          cout << endl;
}

```

1	5	2	k																	
1	1	k																		
2	3	o																		
1	H																			
d	D	d																		
1																				
D	d																			

Az a2 a kivétel miatt nem jött létre, ezért a destruktora sem fog lefutni. (ld. 10. előadás 12-13. dia)!

14. Feladat

Mit ír ki a szabványos kimenetre az alábbi program? Válaszához használja a négyzettrácsos területet!

Ha olyan sorba ír, ahova nem történik kiírás, azért pontot vonunk le!

```

#include <iostream>
using namespace std;
struct Alap {
    Alap(int i = 1)          { cout << i << "k"; }
    virtual void f(int i)   { cout << "f"; }
    void operator=(const Alap& a) { cout << "=" << "f"; }
    virtual ~Alap()        { cout << "d"; }
};
struct Uj :public Alap {
    Uj(int a, const char *m) :Alap(a)    { cout << m << "K"; }
    Uj(const Uj& u) :Alap(u)              { cout << "C" << "f"; }
    void f(int i)                         { cout << "F" << i; }
    ~Uj()                                  { cout << "D"; }
};
void f(Uj u) {
    u.f(2);
}

int main() {
    Uj u12(1, "cde");          cout << endl;
    Alap *pa = &u12;          cout << endl;
    Uj u2 = u12;              cout << endl;
    u12 = u2;                 cout << endl;
    pa->f(1);                  cout << endl;
    f(u12);                   cout << endl;
    Alap a1 = *pa;            cout << endl;
}

```

1	k	c	d	e	K																
C	f																				
=	f																				
F	1																				
C	f	F	2	D	d																
d	D	d	D	d																	

15. Feladat

Mit ír ki a szabványos kimenetre az alábbi program? Válaszához használja a négyzettrácsos területet!

```

#include <iostream>
using namespace std;
struct Alap {
    Alap(int i = 9)          { cout << i << "k"; }
    Alap(const Alap&)        { cout << "c" << "z"; }
    virtual void f(int i)   { cout << "f"; }
    void operator=(const Alap& a) { cout << "=" << "z"; }
    virtual ~Alap()        { cout << "d"; }
};

```

```

};
struct Uj :public Alap {
    Uj(int a = 228, char *m = "z") { cout << m << "K"; }
    Uj(const Uj& u) :Alap(u)      { cout << "C" << "z"; }
    void f(int i)                { cout << "F" << i; }
    ~Uj()                        { cout << "D"; }
};
void f(Uj& u) {
    u.f(9);
}

int main() {
    Uj u19(107, "cde");          cout << endl;
    Alap *pa = new Uj;          cout << endl;
    Uj u2 = u19;                cout << endl;
    u19 = u2;                   cout << endl;
    pa->f(107);                 cout << endl;
    f(u19);                     cout << endl;
    delete pa;                  cout << endl;
}

```

9	k	c	d	e	K				
9	k	z	K						
c	z	C	z						
=	z								
F	1	0	7						
F	9								
D	d								
D	d	D	d						

16. Feladat

Mit ír ki a szabványos kimenetre az alábbi program? Válaszához használja a négyzetrácsos területet!

```

#include <iostream>
using namespace std;
struct A {
    A(int i = 2)                { cout << i << "k"; }
    A(const A&)                 { cout << "c" << "f"; }
    virtual void f(int i)      { cout << "f"; }
    void operator=(const A& a) { cout << "=" << "f"; }
    virtual ~A()               { cout << "d"; }
};
class Uj {
    A a;
public:
    Uj(int a = 83, char *m = "f") { cout << m << "K"; }
    Uj(const Uj& u)                 { cout << "C" << "f"; }
    void f(int i)                  { cout << "F" << i; }
    ~Uj()                           { cout << "D"; }
};
void f(A a) {
    a.f(2);
}

int main() {
    Uj u12(1, "cde");          cout << endl;
    A *pa = new A;            cout << endl;
    Uj u2 = u12;              cout << endl;
    u12 = u2;                 cout << endl;
    pa->f(1);                  cout << endl;
    delete pa;                cout << endl;
}

```

2	k	c	d	e	K				
2	k								
2	k	C	f						
=	f								
F									
d									
D	d	D	d						

17. Feladat

Mit ír ki a szabványos kimenetre az alábbi program? Válaszához használja a négyzettrácsos területet! Abba a sorba írjon, ahol a kiírás keletkezik! Figyeljen a változók élettartamára!

```
#include <iostream>
std::ostream& cout = std::cout;
inline void nl() { cout << std::endl; }

struct Adat {
    int i;
    Adat(int i = -1) :i(i) { cout << i << "k"; }
    Adat(const Adat& a) :i(a.i) { cout << i << "c"; }
    Adat operator+(int n) { Adat tmp(i + n); cout << '+';
return tmp; }
    ~Adat() { cout << "d"; }
};

Adat operator+(int n, Adat& a) { Adat tmp(a + n); return tmp; }

struct Uj {
    Adat a;
    Uj(int a = -3, const char *m = "cde") :a(a) { cout << m << "K"; }
    Uj(const Uj& u) { cout << "C" ; }
    ~Uj() { cout << "D"; }
};

int main() {
    Adat *p = new Adat(200); nl();
    *p = *p + 3; nl();
    *p = 2 + *p; nl();
    Uj u3(4, "ABCDEF"); nl();
    Uj uj16 = u3; nl();
    delete p; nl();
    return 0;
}
```

2	0	0	k																	
2	0	3	k	+	d															
2	0	5	k	+	d															
4	k	A	B	C	D	E	F	K												
-	1	k	C																	
d																				
D	d	D	d																	

A fenti megoldásban nem tüntettük fel a kioptimalizálható másolókonstruktor hívásokat, de ha valaki azt is feltüntette természetesen elfogadtuk.

18. Feladat

Mit ír ki a szabványos kimenetre az alábbi program? Válaszához használja a négyzettrácsos területet! Abba a sorba írjon, ahol a kiírás keletkezik! Figyeljen a változók élettartamára!

```
#include <iostream>
inline std::ostream& Pr() { return std::cout << std::endl; }
template <typename T> std::ostream& Pr(T v) { return std::cout << v; }

struct Adat5 {
    int i;
    Adat5 (int i = -1) :i(i) { Pr(i) << "k"; }
    Adat5 (const Adat5& a) :i(a.i) { Pr(i) << "c"; }
    Adat5& operator+=(int n) { i += n; Pr(i) << "+="; return
*this; }
    virtual ~Adat5 () { Pr("d"); }
};

struct Uj : public Adat5{
    Uj(int a = -5, const char *m = "cde") : Adat5 (a) { Pr(m) << "K"; }
    Uj(const Uj& u) :Adat5 (u) { Pr("C"); }
    ~Uj() { Pr("D"); }
};
```

```
int main() {
    Adat5 a(13);
    a += 3;
    Uj *p = new Uj(4, "ABCDEF");
    Uj uj5 = *p;
    Adat5 *pa = p;
    delete p;
    return 0;
}
```

```
Pr();
Pr();
Pr();
Pr();
Pr();
Pr();
```

1	3	k																	
1	6	+	=																
4	k	A	B	C	D	E	F	K											
4	c	C																	
D	d																		
D	d	d																	

19. Feladat

Mit ír ki a szabványos kimenetre az alábbi program? Válaszához használja a négyzetrácsos területet! Abba a sorba írjon, ahol a kiírás keletkezik! Figyeljen a változók élettartamára!

```
#include <iostream>
struct Pr {
    static int sor;
    int kell;
    Pr(bool kell = 0) :kell(kell) {}
    std::ostream& operator() () { return std::cout; }
    std::ostream& operator[] (int) { return std::cout << sor <<
std::endl; }
    ~Pr() {
        if (kell){ std::cout << std::endl; sor++; }
    }
};
int Pr::sor = 0;

class Adat {
    int a;
public:
    Adat(int i = 3) :a(i) { Pr() () << a << 'k'; }
    Adat(const Adat&) :a(2) { Pr() () << a <<
'c'; }
    Adat& operator=(const Adat& m) { Pr() () << m.a <<
'e'; return *this; }
    ~Adat() { Pr() () << 'd'; }
};

class Ma {
    Adat a;
public:
    Ma(const char *s = "A", int al = 2) :a(al) { Pr() () << s << "K"; }
    ~Ma() { Pr() () << "D"; }
};

int main() {
    static Pr end(3);
    Ma icC("ABCDEF", 3);
    Ma b1 = icC;
    b1 = icC;
    Ma *ma = new Ma[1];
    Pr() [2];
    return 0;
}
```

```
Pr(1);
Pr(1);
Pr(1);
Pr(1);
```

3	k	A	B	C	D	E	F	K											
2	c																		
3	e																		
2	k	A	K																
5																			
D	d	D	d																

20. Feladat

Írja a vonalakra, hogy mit ír ki az alábbi C++ program a standard outputra! Jelölje a szóközőket is! (nem biztos, hogy minden vonalra kell írni)!

```
#include <iostream>
using namespace std;
class Tegla {
    double a;
    double b;
public:
    Tegla(double a = 1, double b = 1)
        :a(a), b(b)      { cout << 'k';}
    Tegla(const Tegla& t)
        :a(t.a), b(t.b) { cout << 'c';}
    void operator=(const Tegla& t)
        { cout << '=';}
    double getA()const  { return a;}
    ~Tegla()            { cout << 'd';}
};
int main() {
    Tegla t1(2, 3);      cout << endl;
    Tegla *pt = new Tegla; cout << endl;
    *pt = t1;           cout << endl;
    cout << pt->getA();  cout << endl;
    delete pt;          cout << endl;
    return 0;
}
```

// _____
 // k _____
 // k _____
 // = _____
 // 1 _____
 // d _____
 // d _____

21. Feladat

Írja a vonalakra, hogy mit ír ki az alábbi C++ program a standard outputra! Jelölje a szóközőket is (nem biztos, hogy minden vonalra kell írni)!

```
#include <iostream>
using namespace std;
class Zh1 {
    char a;
public:
    Zh1() :a('A')          { cout << 'k'; }
    char& operator[] (int) { cout << "x"; return a; }
    const char& operator[] (int) const { cout << "y"; return a; }
    ~Zh1()                { cout << "d"; }
};
int main() {
    Zh1 v1;               cout << endl;
    const Zh1 *pt = new Zh1[2]; cout << endl;
    cout << pt[0][0];      cout << endl;
    delete[] pt;         cout << endl;
    Zh1 v2 = v1;         cout << endl;
    v2[3] = '9';         cout << endl;
    return 0;
}
```

// _____
 // k _____
 // kk _____
 // yA _____
 // dd _____
 // _____
 // x _____
 // dd _____

22. Feladat

Írja a vonalakra, hogy mit ír ki az alábbi C++ program a standard outputra! Jelölje a szóközőket is (nem biztos, hogy minden vonalra kell írni)!

```
#include <iostream>
using namespace std;
```

```

class Valami {
    int a;
public:
    Valami() :a(10)                { cout << 'K'; }
    int& operator[] (int)          { cout << "I"; return a; }
    const int& operator[] (int) const { cout << "J"; return a; }
    ~Valami()                     { cout << "D"; }
};
int main() {
    Valami v1;                    cout << endl;    // _____
    Valami *pt = new Valami[3];   cout << endl;    // KKK _____
    pt[0][1] = 8;                cout << endl;    // I _____
    delete[] pt;                 cout << endl;    // DDD _____
    const Valami v2 = v1;        cout << endl;    // _____
    cout << v2[3];                cout << endl;    // J10 _____
    return 0;                    cout << endl;    // DD _____
}

```

23. Feladat

Írja a vonalakra, hogy mit ír ki az alábbi C++ program a standard outputra! Jelölje a szóközőket is (nem biztos, hogy minden vonalra kell írni)!

```

#include <iostream>
using namespace std;
class Zh1 {
    char a;
public:
    Zh1() :a('A')                { cout << 'K'; }
    char& operator[] (int)       { cout << "I"; return a; }
    const char& operator[] (int) const { cout << "J"; return a; }
    ~Zh1()                       { cout << "D"; }
};
int main() {
    Zh1 v1;                    cout << endl;    // _____
    Zh1 *pt = new Zh1[2];      cout << endl;    // KK _____
    pt[0][0] = '8';           cout << endl;    // I _____
    delete[] pt;              cout << endl;    // DD _____
    const Zh1 v2 = v1;        cout << endl;    // _____
    cout << v2[3];             cout << endl;    // JA _____
    return 0;                 cout << endl;    // DD _____
}

```

24. Feladat

Írja a vonalakra, hogy mit ír ki az alábbi C++ program a standard outputra? Jelölje a szóközőket is! (nem kell minden vonalra írni)

```

#include <iostream>
using namespace std;

int decr(int& r) { return --r;}
int decr(long r) { return --r;}

int main() {
    int var1 = 10;             // _____
    long var2 = decr(var1);   // _____
    cout << var1 << endl;     // 9 _____
    long var3 = decr(var2);   // _____
}

```



```

    cout << var2 << endl;           // 9 _____
    cout << var3 << endl;           // 8 _____
    return 0;                       // _____
}

```

25. Feladat

Írja a vonalakra, hogy az adott sor végrehajtásakor mit ír ki az alábbi C++ program a standard outputra!

Jelölje a szóközőket is! (nem kell minden vonalra írni)

```

#include <iostream>
using namespace std;

int oszt(int a, int b) {
    if (b == 0)
        throw "Nullával oszt";
    return a / b;
}

int main() {
    try {
        cout << oszt(32, 2) << endl;           // _____
        cout << hex << oszt(32, 2) << endl;     // 16 _____
        cout << oszt(32, 0) << endl;           // 10 _____
    } catch (const char *p) {
        cout << "Nono!" << endl;             // _____
    }
    return 0;
}

```

26. Feladat

Írja a vonalakra, hogy az adott sor végrehajtásakor mit ír ki az alábbi C++ program a standard outputra!

Jelölje a szóközőket is! (nem kell minden vonalra írni)

```

#include <iostream>
#include <iomanip>
namespace ronda {
    const unsigned int N = 5;
    const unsigned int M = 10;
    int adat[N*M];
    int& vec(unsigned int i) {
        return adat[i];
    }
    int& vec(unsigned int i, unsigned int j) {
        return adat[i*M+j];
    }
}
using namespace std;
using ronda::vec;

int main() {
    vec(10) = 4;
    vec(1,0)++;
    cout << vec(10) << endl;           // 5 _____
    vec(2) = 2;                       // _____
    cout << ++vec(2) << endl;           // 3 _____
    cout << setw(3) << vec(2) << endl;   // 003 _____
    return 0;
}

```

}

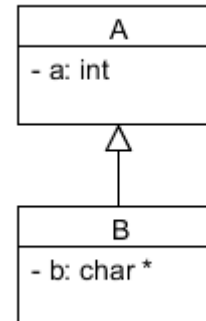
27. Feladat

- a. **Rajzolja** le az alábbi programban deklarált **A** és **B** osztályok kapcsolatát! Jelölje tagváltozók típusát és láthatóságát, de a tagfüggvényeket most **ne** jelölje, hogy a megadott helyen elférjen a rajz!

```
#include <iostream>
using namespace std;
class A {
    int a;
public:
    A(int i) :a(i)
        { cout << "A ctor " << a; }
    A(const A& ar) :a(ar.a)
        { cout << "A copy "; }
    void operator=(const A& ar)
        { a = ar.a; cout << "A assign"; }
    virtual void print(ostream& os) const
        { os << a; }
    virtual ~A() { cout << "A dtor "; }
};
class B : public A {
    char *b;
public:
    B(int i = 0, char* d = "B0") :A(i), b(d) { cout << " B ctor " << d; }
    B(const B& br) :A(br), b(br.b) { cout << " B copy "; }
    void print(ostream& os) const { os << b; }
    ~B() { cout << "B dtor "; }
};

int main() {
    A a1(1);
    B b1;
    B b2(2, "B2");
    B b3 = b2;
    b1 = b3;
    A *pa = &b1;
    pa->A::print(cout);
    pa->print(cout);
    return 0;
}
```

Ide rajzoljon:



- b. **Írja a vonalakra**, hogy mit ír ki a fenti C++ program a standard outputra az egyes utasítások végrehajtásakor! Jelölje a szóközöket is! (lehet, hogy nem kell minden vonalra írni.)

- c. Egészítse ki a fenti programot úgy, hogy az alábbi kódrészlet a megjegyzésben leírtaknak megfelelően működjön! A tagváltozó láthatóságát ne változtassa meg! (A standard outputon bármi keletkezhet!)

```
cerr << A(12); // 12-t ír a standard hibakimenetre
cerr << A(234); // 234-et ír a standard hibakimenetre
cerr << A(0) << endl; // 0-t, majd egy soremelést ír a
// standard hibakimenetre
```

```
ostream& operator<<(ostream &os, const A& a) {
    a.print(os);
    return os;
}
```

28. Feladat

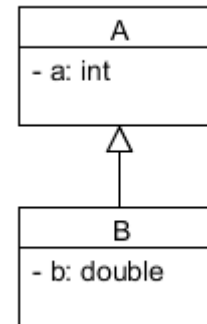
- a. **Rajzolja** le az alábbi programban deklarált **A** és **B** osztályok kapcsolatát! Jelölje tagváltozók típusát és láthatóságát, de a tagfüggvényeket most **ne** jelölje, hogy a megadott helyen elférjen a rajz!

```
#include <iostream>
using namespace std;
class A {
    int a;
public:
    A(int i = 0) :a(i)
    { cout << "A ctor " << a; }
    void operator=(const A& ar)
    { a = ar.a; cout << "A assign"; }
    virtual void write(ostream& os) const
    { os << a; }
    virtual ~A()
    { cout << "A dtor "; }
};
```

```
class B : public A {
    double b;
public:
    B(int i = 0, double d = 0) :A(i), b(d) { cout << " B ctor " << d; }
    B(const B& br) :A(br), b(br.b){ cout << " B copy "; }
    void write(ostream& os) const { os << b;}
    ~B() { cout << "B dtor "; }
};
```

```
int main() {
    B b1(1);          cout << endl; // A°ctor°1 B°ctor°0
    B b2(3, 3.2);    cout << endl; // A°ctor°2 B°ctor°3.2
    b1 = b2;         cout << endl; // A°assign
    A *pa = new B(6, 4.4); cout << endl; // A°ctor°6 B°ctor°4.4
    pa->A::write(cout); cout << endl; // 6
    pa->write(cout);  cout << endl; // 4.4
    delete pa;       cout << endl; // B°dtor°A°dtor°
} // B°dtor°A°dtor°B°dtor°A°dtor°
```

Ide rajzoljon:



- b. **Írja a vonalakra**, hogy mit ír ki a fenti C++ program a standard outputra az egyes utasítások végrehajtásakor! Jelölje a szóközőket is! (lehet, hogy nem kell minden vonalra írni.)
- c. Egészítse ki a fenti programot úgy, hogy az alábbi kódrészlet a megjegyzésben leírtaknak megfelelően működjön! A tagváltozó láthatóságát ne változtassa meg! (A standard outputon bármi keletkezhet!)

```
cerr << A(12); // 12-t ír a standard hibakimenetre
cerr << A(234); // 234-et ír a standard hibakimenetre
cerr << A() << endl; // 0-t, majd egy soremelést ír a
// standard hibakimenetre
```



```
ostream& operator<<(ostream &os, const A& a) {
    a.print(os);
    return os;
}
```

29. Feladat

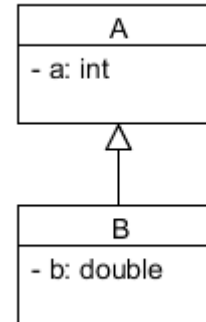
- a. **Rajzolja** le az alábbi programban deklarált **A** és **B** osztályok kapcsolatát! Jelölje tagváltozók típusát és láthatóságát, de a tagfüggvényeket most **ne** jelölje, hogy a megadott helyen elférjen a rajz!

```
#include <iostream>
using namespace std;
class A {
    int a;
public:
    A(int i = 0) :a(i)
    { cout << "A ctor " << a; }
    A(const A& ar) :a(ar.a)
    { cout << "A copy "; }
    void operator=(const A& ar)
    { a = ar.a; cout << "A assign"; }
    void write(ostream& os) const
    { os << a; }
    virtual ~A()
    { cout << "A dtor "; }
};
```

```
class B : public A {
    double b;
public:
    B(int i = 0, double d = 0) :A(i), b(d) { cout << " B ctor " << d; }
    void write(ostream& os) const { os << b; }
    ~B()
    { cout << "B dtor "; }
};
```

```
int main() {
    A a1(1);          cout << endl; // _____
    B b2(3, 3.2);    cout << endl; // A°ctor°3°B°ctor°3.2 _____
    B b3 = b2;      cout << endl; // A°copy _____
    A *pa = new B(6, 4.4); cout << endl; // A°ctor°6°B°ctor°4.4 _____
    pa->A::write(cout); cout << endl; // 6 _____
    pa->write(cout);  cout << endl; // 6 _____
    delete pa;       cout << endl; // B°dtor°A°dtor° _____
} // B°dtor°A°dtor°B°dtor°A°dtor°A°dtor° _____
```

Ide rajzoljon:



- b. **Írja a vonalakra**, hogy mit ír ki a fenti C++ program a standard outputra az egyes utasítások végrehajtásakor! Jelölje a szóközöket is! (lehet, hogy nem kell minden vonalra írni.)
- c. Egészítse ki a fenti programot úgy, hogy az alábbi kódrészlet a megjegyzésben leírtaknak megfelelően működjön! A tagváltozó láthatóságát ne változtassa meg! (A standard outputon bármi keletkezhet!)

```
cerr << A(12); // 12-t ír a standard hibakimenetre
cerr << A(234); // 234-et ír a standard hibakimenetre
cerr << A() << endl; // 0-t, majd egy soremelést ír a
// standardhibakimenetre
```

```
ostream& operator<<(ostream &os, const A& a) {
    a.print(os);
    return os;
}
```

30. Feladat

Írja a vonalakra, hogy mit ír ki az alábbi C++ program a standard outputra? Jelölje a szóközőket is! (nem kell minden vonalra írni)

```
#include <iostream>
using namespace std;

inline int novel(int& r) { return r++;}
inline int novel(short r) { return ++r;}

int main() {
    int var1 = 16; // _____
    short var2 = novel(var1); // _____
    int var3 = novel(var2); // _____
    cout << var1 << endl; // 17 _____
    cout << var2 << endl; // 16 _____
    cout << var3 << endl; // 17 _____
    return 0; // _____
}
```

31. Feladat

Írja a vonalakra, hogy mit ír ki az alábbi C++ program a standard outputra? Jelölje a szóközőket is! (nem kell minden vonalra írni)

```
#include <iostream>
using namespace std;

int csokkent(int &a) {
    if (a <= 0)
        throw "Nulla";
    return --a;
}

int main() {
    try {
        int a = 2; // _____
        cout << csokkent(a) << endl; // 1 _____
        cout << csokkent(a) << endl; // 0 _____
        cout << csokkent(a) << endl; // _____
    } catch (const char *p) {
        cout << "Nono!" << endl; // Nono _____
    }
    return 0; // _____
}
```

32. Feladat

Írja a vonalakra, hogy mit ír ki az alábbi C++ program a standard outputra? Jelölje a szóközöket is! (nem kell minden vonalra írni)

```
#include <iostream>
using namespace std;

double reciprok(int a) {
    if (a == 0)
        throw "Nulla";
    return 1.0/a;
}

int main() {
    try {
        cout << reciprok(2) << endl;           // 0.5
        cout << reciprok(1) << endl;           // 1
        cout << reciprok(0) << endl;           //
    } catch (const char *p) {                 //
        cout << "Jaj!" << endl;               // Jaj!
    }                                          //
    return 0;                                 //
}
```

33. Feladat

Írja a vonalakra, hogy mit ír ki az alábbi C++ program a standard outputra? Jelölje a szóközöket is! (nem kell minden vonalra írni)

```
#include <iostream>
using namespace std;

inline int csokk(short r) { return --r;}
inline int csokk(int& r)  { return r--;}

int main() {
    int var1 = 200;           //
    short var2 = csokk(var1); //
    int var3 = csokk(var2);   //
    cout << var1 << endl;      // 199
    cout << var2 << endl;      // 200
    cout << var3 << endl;      // 199
    return 0;                 //
}
```

34. Feladat

Írja a vonalakra, hogy mit ír ki az alábbi C++ program a standard outputra? Jelölje a szóközöket is! (nem kell minden vonalra írni)

```
#include <iostream>
using namespace std;

double reciprok(int a) {
    if (a == 0)
        throw "Nulla";
    return 1.0/a;
}
```

```

}

int main() {
    try {
        cout << reciprok(2) << endl;           // 0.5
        cout << reciprok(1) << endl;           // 1
        cout << reciprok(0) << endl;           //
    } catch (const char *p) {                 //
        cout << "Jaj!" << endl;               // Jaj!
    }                                          //
    return 0;                                 //
}

```

35. Feladat

Írja a vonalakra, hogy mit ír ki az alábbi C++ program a standard outputra? Jelölje a szóközőket is! (nem kell minden vonalra írni)

```

#include <iostream>
using namespace std;
inline int fv(int a) { return a + a; }
namespace saját {
    int var1;
    inline int fv(int a) {
        var1 = 2*a;
        return a*a;
    }
}
int main() {
    int var1 = fv(3);           //
    cout << var1 << endl;       // 6
    cout << saját::fv(var1) << endl; // 36
    cout << saját::var1 << endl; // 12
    return 0;                   //
}

```

36. Feladat

Írja a vonalakra, hogy mit ír ki az alábbi C++ program a standard outputra? Jelölje a szóközőket is! (nem kell minden vonalra írni)

```

#include <iostream>
using std::ostream;
using std::cout;

class A {
    int a;
public:
    A(int a = 0) :a(a)           { cout << "K" << a;}
    A(const A& a1):a(a1.a)       { cout << "C" << a;}
    A& operator=(const A& rhs)  { a = rhs.a; cout << "=" << a;
                                return *this; }
    int get() const              { return a; }
    void f(A a0)                  { cout << "f" << a0; }
    friend ostream& operator<<(ostream& os, const A& rhs);
    ostream& operator<<(ostream& os) const;
    ~A()                          { cout << "D"; }
};

```



```

ostream& operator<<(ostream& os, const A& rhs) {
    return os << "G" << rhs.a;
}

ostream& A::operator<<(ostream& os) const {
    return os << "L" << a;
}

void endl() { cout << std::endl; }

int main() {
    A a;          endl(); // K0
    A b = a;     endl(); // C0
    A c(12);     endl(); // K12
    a = c;       endl(); // =12
    c.f(b);     endl(); // C0fG0D
    cout << a;  endl(); // G12
    a << cout;  endl(); // L12
    return 0;    // DDD
}

```

Az **A** osztály **módosítása nélkül** készítsen egy olyan ***** operátort, amellyel két ilyen objektum összeszorozható! Ne dobjon kivételt az alábbi kódrészlet!

```

A a5(5), a6(6), a7(7);
if ((a5 * a6).get() != 30 ||
    (a5 * a7).get() != 35) throw("HIBA");

```

```

A operator*(const A& lhs, const A& rhs) {
    return A(lhs.get()*rhs.get());
}

```

37. Feladat

- a. Írja a vonalakra, hogy mit ír ki az alábbi C++ program a standard outputra? Jelölje a szóközőket is! (nem kell minden vonalra írni)

```

#include <iostream>
using std::ostream;
using std::cout;

class B {
    int b;
public:
    B(int b = 10) :b(b)          { cout << "K" << b; }
    B(const B& b0) :b(b0.b)     { cout << "C" << b; }
    B& operator=(const B& rhs) { b = rhs.b; cout << "=" << b;
                                return *this; }

    int get() const             { return b; }
    void g(B& b1)                { cout << "f" << b1.b; }
    friend ostream& operator<<(ostream&, const B);
    ostream& operator<<(ostream&) const;
    ~B()                          { cout << "D"; }
};

ostream& operator<<(ostream& os, const B rhs) {
    return os << "G" << rhs.b;
}

ostream& B::operator<<(ostream& os) const {
    return os << "L" << b;
}

```

```

}
void endl() { cout << std::endl; }
int main() { // _____
    B a;      endl(); // K10 _____
    B b = a;  endl(); // C10 _____
    B c(1);   endl(); // K1  _____
    b = c;    endl(); // =1  _____
    c.g(b);   endl(); // f1  _____
    cout << a; endl(); // C10G10D _____
    a << cout; endl(); // L10 _____
    return 0; // DDD _____
}

```

- b. A B osztály módosítása nélkül készítsen egy olyan + operátort, amellyel két ilyen objektum összeadható! Ne dobjon kivételt az alábbi kódrészlet!

```

B a5(5), a6(6), a7(7);
if ((a5 + a6).get() != 11 ||
    (a5 + a7).get() != 12) throw("HIBA");

```

```

B operator+(const B& lhs, const B& rhs) {
    return B(lhs.get()+rhs.get());
}

```

38. Feladat

- a. Írja a vonalakra, hogy mit ír ki az alábbi C++ program a standard outputra? Jelölje a szóközőket is! (nem kell minden vonalra írni)

```

#include <iostream>
using namespace std;

class A {
    int a;
public:
    A(int a = 0) :a(a) { cout << "K" << a; }
    A(const A& a1):a(a1.a) { cout << "C" << a; }
    A& operator=(int i) { a = i; cout << "=" << i;
        return *this; }

    int get() const { return a; }
    void f(A a0) { a0 << cout << "f"; }
    friend ostream& operator<<(ostream& os, const A& rhs);
    ostream& operator<<(ostream& os) const;
    ~A() { cout << "D"; }
};

ostream& operator<<(ostream& os, const A& rhs) {
    return os << "G" << rhs.a;
}

ostream& A::operator<<(ostream& os) const {
    return os << "L" << a;
}

inline void nl() { cout << std::endl; }

int main() { // _____
    A a;      nl(); // K0  _____
    A b = a;  nl(); // C0  _____
    A c(12);  nl(); // K12 _____
}

```

```

a = 8;      nl();      // =8 _____
c.f(b);    nl();      // C0L0fD _____
cout << a;  nl();      // G8 _____
a << cout;  nl();      // L8 _____
return 0;   // DDD _____
}

```

- b. Az **A** osztály **módosítása nélkül** készítsen egy olyan **- operátort**, amellyel két ilyen objektum különbsége kiszámolható! Ne dobjon kivételt az alábbi kódrészlet!

```

A a7(7), a6(6), a5(5);
if ((a7 - a6).get() != 1 ||
    (a7 - a5).get() != 2) throw("HIBA");
A operator-(const A& lhs, const A& rhs) {
    return A(lhs.get()-rhs.get());
}

```

39. Feladat

- a. Írja a vonalakra, hogy mit ír ki az alábbi C++ program a standard outputra? Jelölje a szóközőket is! (nem kell minden vonalra írni)

```

#include <iostream>
using namespace std;

class B {
    int b;
public:
    B(int b = 9) :b(b)          { cout << "K" << b;}
    B(const B& b0):b(b0.b)     { cout << "C" << b;}
    B& operator=(int i)       { b = i; cout << "=" << b;
                               return *this; }
    int get() const           { return b; }
    void g(B& b1)             { b1 << cout << "f"; }
    friend ostream& operator<<(ostream&, const B);
    ostream& operator<<(ostream&) const;
    ~B()                      { cout << "D"; }
};

ostream& operator<<(ostream& os, const B rhs) {
    return os << "G" << rhs.b;
}

ostream& B::operator<<(ostream& os) const {
    return os << "L" << b;
}

void nl() { cout << std::endl; }

int main() {
    B a;      nl();      // K9 _____
    B b = a;  nl();      // C9 _____
    B c(1);   nl();      // K1 _____
    b = 12;   nl();      // =12 _____
    c.g(b);   nl();      // L12f _____
    cout << a; nl();      // C9G9D _____
    a << cout; nl();      // L9 _____
    return 0; // DDD _____
}

```

- b. A B osztály módosítása nélkül készítsen olyan összehasonlító operátort (<), amellyel két ilyen objektum a tárolt egész alapján összehasonlítható! Ne dobjon kivételt az alábbi kódrészlet!

```
B a5(5), a6(6);
if ((a5 < a6) != true || (a6 < a5) != false) throw("HIBA");
```

```
bool operator<(const B& lhs, const B& rhs) {
    return lhs.get() < rhs.get();
```

40. Feladat

- a. Írja a vonalakra, hogy mit ír ki az alábbi C++ program a standard outputra? Jelölje a szóközőket is! (nem kell minden vonalra írni)

```
#include <iostream>
using namespace std;

class A {
    int a;
public:
    A(int a = 6) :a(a)           { cout << "K" << a;}
    A(const A& a1):a(a1.a)      { cout << "C" << a;}
    A& operator=(int i)        { a = i; cout << "=" << i;
                                return *this; }
    int get() const            { return a; }
    void f(A a0)                { a0 << cout << "f"; }
    friend ostream& operator<<(ostream& os, const A& rhs);
    ostream& operator<<(ostream& os) const;
    ~A()                        { cout << "D"; }
};

ostream& operator<<(ostream& os, const A& rhs) {
    return os << "G" << rhs.a;
}

ostream& A::operator<<(ostream& os) const {
    return os << "L" << a;
}

inline void nl() { cout << std::endl; }

int main() {
    A a;           nl(); // _____
    A b = a;      nl(); // K6 _____
    A c(30);      nl(); // C6 _____
    a = 48;       nl(); // K30 _____
    a = 48;       nl(); // =48 _____
    c.f(b);      nl(); // C6L6fD _____
    cout << a;    nl(); // G48 _____
    a << cout;    nl(); // L48 _____
    return 0;     nl(); // DDD _____
}
```

- b. Az A osztály módosítása nélkül készítsen egy olyan *= operátort, amellyel egy ilyen objektum megszorozható egy egész számmal! Ne dobjon kivételt az alábbi kódrészlet!

```
A a5(5), a6(6);
if ((a5 *= 6).get() != 30 || (a5.get() != 30) ||
    (a6 *= 8).get() != 48) throw("HIBA");
```

```
A& operator*=(A& lhs, int i) {
```

```
return lhs = lhs.get() * i;
}
```

41. Feladat

- a. Írja a vonalakra, hogy mit ír ki az alábbi C++ program a standard outputra? Jelölje a szóközőket is! (nem kell minden vonalra írni)

```
#include <iostream>
using namespace std;

class B {
    int b;
public:
    B(int b = 19) :b(b)           { cout << "K" << b;}
    B(const B& b0) :b(b0.b)      { cout << "C" << b;}
    B& operator=(int i)         { b = i; cout << "=" << b;
                                return *this; }

    int get() const             { return b; }
    void g(B& b1)                { b1 << cout << "f"; }
    friend ostream& operator<<(ostream&, const B);
    ostream& operator<<(ostream&) const;
    ~B()                          { cout << "D"; }
};

ostream& operator<<(ostream& os, const B rhs) {
    return os << "G" << rhs.b;
}

ostream& B::operator<<(ostream& os) const {
    return os << "L" << b;
}

void nl() { cout << std::endl; }

int main() { //
    B a;      nl(); // K19
    B b = a;  nl(); // C19
    B c(1);   nl(); // K1
    b = 14;   nl(); // =14
    c.g(b);   nl(); // L14f
    cout << a; nl(); // C19G19D
    a << cout; nl(); // L19
    return 0; // DDD
}
```

- b. A B osztály módosítása nélkül készítsen egy olyan += operátort, amellyel egy ilyen objektumhoz hozzá lehet adni egy egész számot! Ne dobjon kivételt az alábbi kódrészlet!

```
B a5(5), a6(6);
if ((a5 += 6).get() != 11 || (a5.get() != 11) ||
    (a6 += 8).get() != 14) throw("HIBA");

B& operator+=(B& lhs, int i) {
    return lhs = lhs.get() + i;
}
```

42. Feladat

A pontozott helyek felhasználásával egészítse ki az alábbi programot úgy, hogy a főprogram kommentezett sorai az előre megadott kimenetet (1,2 ill. 1) adják!

Ügyeljen a tagváltozók elérhetőségére!

Ezt követően a főprogram minden sorához is adja meg, hogy az mit ír ki a szabványos kimenetre! Jelölje a szóköz karaktereket is! (Nem biztos, hogy minden megadott helye kell írnia!)

```
#include <iostream>
using namespace std;

class Pont1 {
    int x;          // x koordináta
public:
    Pont1(int x = 0, int y = 0) :x(x)          { cout << "P1ctor ";}
    Pont1(const Pont1& p) :x(p.x)            { cout << "P1copy ";}
    int get() const { return x; }
    Pont1& operator=(const Pont1& p) { x = p.x; cout << "P1= "; return
*this; }
    ~Pont1()                                  { cout << "P1dctor "; }
};

class Pont2 {
    Pont1 p1;
    int y;          // y koordináta
public:
    Pont2(int x = 0, int y = 0)
        : p1(x), y(y) /*.....*/          { cout << "P2ctor "; }
    Pont2(const Pont2& p2)
        : p1(p2.getx()), y(p2.y) /*....*/  { cout << "P2copy "; }
    int getx() const { return p1.get(); }
    int gety() const { return y; }
    ~Pont2()                                  { cout << "P2dctor "; }
};
/* ..... */
ostream& operator<<(ostream& os, const Pont2& p2) {
    os << p2.getx() << ', ' << p2.gety();
    return os;
}
/* ..... */

int main() {
    Pont1 p1(1);          cout << endl; // P1ctor _____
    Pont2 p2(p1.get(), 2); cout << endl; // P1ctor P2ctor _____
    Pont2 p = p2;        cout << endl; // P1ctor P2copy _____
    cout << p             << endl;    // 1,2 _____
    Pont2 *pp = new Pont2(p); cout << endl; // P1ctor P2copy _____
    cout << pp->getx()    << endl;    // 1 _____
    delete pp;          cout << endl; // P2dctor p1dctor _____
    return 0;          // P2dctor P1dctor P2dctor P1dctor P1dctor _____
}
```

43. Feladat

A **pontozott** helyek felhasználásával **egészítse ki** az alábbi programot úgy, hogy a főprogram kommentezett sorai az **előre megadott** kimenetet (**10 P2= 20,30**) adják!

Ügyeljen a tagváltozók elérhetőségére!

Ezt követően a főprogram minden sorához is adja meg, hogy az mit ír ki a szabványos kimenetre! Jelölje a szóköz karaktereket is! (Nem biztos, hogy minden megadott helyre írnia kell!)

Egy lehetséges megoldás:

```
#include <iostream>
using namespace std;
class Pont1 {
    int x;          // x koordináta
public:
    Pont1(int x = 0)
        : x(x) { cout << "P1ctor "; }
    Pont1(const Pont1& p)
        : x(p.x) { cout << "P1copy "; }
    Pont1& operator=(const Pont1& p1)
        { x = p1.x; /*.....*/ return *this; }
    virtual void print(ostream& os) const { os << x; }
    virtual ~Pont1() { cout << "P1dctor "; }
};
class Pont2 :public Pont1 {
    int y;          // y koordináta
public:
    Pont2(int x = 0, int y = 0)
        :Pont1(x), y(y) { cout << "P2ctor "; }
    Pont2& operator=(const Pont2& p2)
        { Pont1::operator=(p2); y = p2.y; cout << "P2= "; return
*this; }
    ~Pont2() { cout << "P2dctor "; }
    /*..... */
    void print(ostream& os) const {
        Pont1::print(os);
        os << ", " << y;
    }
    /*..... */
};
/*..... */
ostream& operator<<(ostream& os, const Pont1& p) {
    p.print(os);
    return os;
}
/*..... */

int main() {
    Pont1 p1(10);    cout << endl; // P1ctor
    cout << p1      << endl; // 10
    Pont2 p23(20,30); cout << endl; // P1ctor P2ctor
    Pont2 p20;      cout << endl; // P1ctor P2ctor
    p20 = p23;      cout << endl; // P2=
    p20.print(cout); cout << endl; // 20,30
    Pont1 *pp = &p20; cout << endl; //
    pp->print(cout); cout << endl; // 20,30
    return 0;      // P2dctor P1dctor P2dctor P1dctor
P1dctor
}
```

}

44. Feladat

A **pontozott** helyek felhasználásával **egészítse ki** az alábbi programot úgy, hogy a főprogram kommentezett sorai az **előre megadott** kimenetet (1,2 ill. 4,5,6) adják!

Ügyeljen a tagváltozók elérhetőségére!

Ezt követően a főprogram minden sorához is adja meg, hogy az mit ír ki a szabványos kimenetre! Jelölje a szóköz karaktereket is! (Nem biztos, hogy minden megadott helye kell írnia!)

```
#include <iostream>
using namespace std;
class Pont {
    int x;          // x koordináta
    int y;          // y koordináta
public:
    Pont(int x = 0, int y = 0)
        : x(x), y(y)                    { cout << "Pctor "; }
    Pont(const Pont& p)
        : x(p.x), y(p.y)                { cout << "Pcopy "; }
    virtual void print(ostream& os) const { os << x << ", " << y; }
    virtual ~Pont()                       { cout << "Pdtor "; }
};
```

```
class Pont3D :public Pont {
    int z;          // z koordináta
public:
    Pont3D(int x = 0, int y = 0, int z = 0)
        :Pont(x, y), z(z)                { cout << "P3ctor "; }
    Pont3D(const Pont3D& p3d)
        :Pont(p3d), z(p3d.z)             { cout << "P3copy "; }
    ~Pont3D()                             { cout << "P3dtor "; }
    /* ..... */
    void print(ostream& os) const {
        Pont::print(os);
        os << ", " << z;
    }
    /* ..... */
};
/* ..... */
ostream& operator<<(ostream& os, const Pont& p) {
    p.print(os);
    return os;
}
/* ..... */
```

```
int main() {
    Pont v12(1,2);    cout << endl;        // Pctor _____
    Pont v = v12;    cout << endl;        // Pcopy _____
    cout << v12 << endl;                    // 1,2 _____
    Pont* vp[5];     cout << endl;        // _____
    vp[0] = new Pont3D(4,5,6); cout<<endl; // Pctor P3ctor _____
    vp[0]->print(cout); cerr << endl;    // 4,5,6 _____
    delete vp[0];    cout << endl;        // P3dtor Pdtor _____
    return 0;        // Pdtor Pdtor _____
}
```


}

45. Feladat

A **pontozott** helyek felhasználásával **egészítse ki** az alábbi programot úgy, hogy a főprogram kommentezett sorai az **előre megadott** kimenetet (1 2,3 2) adják!

Ügyeljen a tagváltozók elérhetőségére!

Ezt követően a főprogram minden sorához is adja meg, hogy az mit ír ki a szabványos kimenetre! Jelölje a szóköz karaktereket is! (Nem biztos, hogy minden megadott helyre írnia kell!)

```
#include <iostream>
using namespace std;
class Pont1 {
    int x;          // x koordináta
public:
    Pont1(int x = 0)
        : x(x) { cout << "P1ctor "; }
    Pont1(const Pont1& p)
        : x(p.x) { cout << "P1copy "; }
    /*.....*/ void print(ostream& os) const { os << x; }
    virtual ~Pont1() { cout << "P1dctor "; }
};
class Pont2 :public Pont1 {
    int y;          // y koordináta
public:
    Pont2(int x = 0, int y = 0)
        :Pont1(x), y(y) { cout << "P2ctor "; }
    Pont2& operator=(const Pont2& p2)
        { Pont1::operator=(p2); y = p2.y; cout << "P2= "; return
*this;}
    ~Pont2() { cout << "P2dctor "; }
    /*.....*/
    void print(ostream& os) const {
        Pont1::print(os);
        os << "," << y;
    }
    /*.....*/
};
/*.....*/
ostream& operator<<(ostream& os, const Pont1& p) {
    p.print(os);
    return os;
}
/*.....*/

int main() {
    Pont1 p1(1);      cout << endl; // P1ctor _____
    cout << p1        << endl; // 1 _____
    Pont2 p23(2,3);  cout << endl; // P1ctor P2ctor _____
    Pont2 p20;       cout << endl; // p1ctor P2ctor _____
    p20 = p23;       cout << endl; // P2= _____
    p20.print(cout); cout << endl; // 2,3 _____
    Pont1 *pp = &p20; cout << endl; // _____
    pp->print(cout); cout << endl; // 2 _____
    return 0;        // P2dctor P1dctor P2dctor P1dctor P1dctor _____
}
```

}

46. Feladat

Írja a vonalakra, hogy mit ír ki az alábbi C++ program a standard outputra! Jelölje a szóközőket is (nem biztos, hogy minden vonalra kell írni)!

```
#include <iostream>
using namespace std;
class Zh1 {
    char a;
public:
    Zh1() :a('A')           { cout << 'k'; }
    Zh1(const Zh1& zh) :a(zh.a) { cout << 'c'; }
    Zh1& operator=(const Zh1& zh) { cout << '='; a = zh.a;
                                   return *this; }
    ~Zh1()                 { cout << 'd'; }
};
int main() {
    Zh1 v1;                cout << endl; // k _____
    Zh1 v2 = v1;          cout << endl; // c _____
    Zh1 v3[2];            cout << endl; // kk _____
    v3[0] = v2;          cout << endl; // = _____
    return 0;            cout << endl; // dddd _____
}
```

47. Feladat

Írja a vonalakra, hogy mit ír ki az alábbi C++ program a standard outputra! Jelölje a szóközőket is (nem biztos, hogy minden vonalra kell írni)!

```
#include <iostream>
using namespace std;
class Zh1 {
    char a;
public:
    Zh1() :a('A')           { cout << 'k'; }
    Zh1(const Zh1& zh) :a(zh.a) { cout << 'c'; }
    ~Zh1()                 { cout << 'd'; }
};
ostream& operator<<(ostream& os, const Zh1&) { os << 'o'; return os;}
int main() {
    Zh1 v1;                cout << endl; // k _____
    Zh1 v2 = v1;          cout << endl; // c _____
    Zh1 v3[2];            cout << endl; // kk _____
    cout << v1 << endl;    // o _____
    return 0;            cout << endl; // dddd _____
}
```

48. Feladat

Írja a vonalakra, hogy mit ír ki az alábbi C++ program a standard outputra! Jelölje a szóközőket is (nem biztos, hogy minden vonalra kell írni)!

```
#include <iostream>
using namespace std;
class Zh1 {
    char a;
public:
    Zh1() :a('A')           { cout << 'k'; }
    Zh1(const Zh1& zh) :a(zh.a) { cout << 'c'; }
```

```

    ~Zh1() { cout << 'd'; }
};
ostream& operator<<(ostream& os, const Zh1&) { os << 'o'; return os;}
int main() {
    Zh1 v1;          cout << endl;
    Zh1 v2 = v1;    cout << endl;
    Zh1 v3[2];      cout << endl;
    cout << v1 << endl;
    return 0;
}

```

// _____
 // k _____
 // c _____
 // kk _____
 // o _____
 // dddd _____

49. Feladat

Írja a vonalakra, hogy mit ír ki az alábbi C++ program a standard outputra? Jelölje a szóközőket is! (nem kell minden vonalra írni)

```

#include <iostream>
using namespace std;

inline void set(int& t, int v = 1) { t = v; }
inline int sum(const int& t, int c = 1) { return t + c; }
inline long sum(long t, int c = 0) { return t - c; }

int main() {
    int t1[10];
    long t2[10] = { 10 };
    set(t1[0]);
    set(t1[1], 10);
    cout << sum(t1[0]) << endl;
    cout << oct << sum(t2[0]) << endl;
    cout << hex << sum(t1[1], 6) << endl;
    return 0;
}

```

// _____
 // _____
 // _____
 // _____
 // 2 _____
 // 12 _____
 // 10 _____
 // _____

50. Feladat

Írja a vonalakra, hogy mit ír ki az alábbi C++ program a standard outputra? Jelölje a szóközőket is! (nem kell minden vonalra írni)

```

#include <iostream>
using namespace std;

void set(int& t, int v = 0) { t = v; }
int sum(const int& t, int c = 10) { return t + c; }
long sum(long t, int c = 0) { return t - c; }

int main() {
    int t1[10];
    long t2[10] = { 10 };
    set(t1[0]);
    set(t1[1], 10);
    cout << sum(t1[0]) << endl;
    cout << oct << sum(t2[0]) << endl;
    cout << hex << sum(t1[1], 6) << endl;
    return 0;
}

```

// _____
 // _____
 // _____
 // _____
 // 10 _____
 // 12 _____
 // 10 _____
 // _____

51. Feladat

Írja a vonalakra, hogy mit ír ki az alábbi C++ program a standard outputra? Jelölje a szóközőket is! (nem kell minden vonalra írni)

```
#include <iostream>
using namespace std;

void set(int& t, int v = 0) { t = v; }
int sum(const int& t, int c = 6) { return t + c; }
long sum(long t, int c = 0) { return t - c; }

int main() {
    int t1[10]; // _____
    long t2[10] = { 10 }; // _____
    set(t1[0]); // _____
    set(t1[1], 10); // _____
    cout << sum(t1[0]) << endl; // 6
    cout << oct << sum(t2[0]) << endl; // 12
    cout << hex << sum(t1[1], 6) << endl; // 10
    return 0; // _____
}
```

52. Feladat

Írja a vonalakra, hogy mit ír ki az alábbi C++ program a standard outputra? Jelölje a szóközőket is! (nem kell minden vonalra írni)

```
#include <iostream>
using namespace std;

void set(int& t, int v = 0) { t = v; }
int sum(const int& t, int c = 8) { return t + c; }
long sum(long t, int c = 0) { return t - c; }

int main() {
    int t1[10]; // _____
    long t2[10] = { 10 }; // _____
    set(t1[0]); // _____
    set(t1[1], 10); // _____
    cout << sum(t1[0]) << endl; // 8
    cout << oct << sum(t2[0]) << endl; // 12
    cout << hex << sum(t1[1], 6) << endl; // 10
    return 0; // _____
}
```

53. Feladat

Racionális számokat egy struktúrában úgy tárolunk, hogy egész számként tároljuk a számlálót és a nevezőt:

```
struct Rac { int szamlalo; int nevezo; };
```

- a. **Készítsen** C++ nyelven a számok kiírásához olyan függvényt, amivel azok kiírhatók a megszokott módon egy ostream típusú objektumra. Működjön helyesen az alábbi kódrészlet:

```
Rac r1 = { 3, 4 }, r2 = { 8, 9 };
std::cout << "r1:" << r1 << " r2:" << r2 << std::endl; //r1:3/4 r2:8/9
```

Úgy írja ki a két értéket, hogy a számláló és a nevező közé tegyen valamilyen nyomtatható elválasztó karaktert (pl. 3/4, vagy 3:4, stb.)!

- b. Ha elkészült, írja a fenti utasítás után a vonalra, hogy megvalósítása szerint mi jelenik meg a standard kimeneten az utasítás hatására!

Egy lehetséges megoldás:

```
std::ostream& operator<<(std::ostream& os, const Rac& r) {  
    return os << r.szamlalo << "/" << r.nevezo;  
}
```