20

YIGIRMANCHI MA’RUZA

Dasturlash 1 · C++ · 14-mavzu

Operatorlarni qayta yuklash

A’zo yoki erkin funksiya · += orqali + · taqqoslash va <=> · oqim operatorlari · [], (), ->, bool · KattaSon loyihasi

operator+=

<=>

operator<<

operator[]

ISINISH

Nima uchun `a + b` yozish `a.qosh(b)` dan qulayroq?

Matematikada biz belgilar bilan fikrlaymiz. O‘z sinflarimiz ham shunday “gapira” oladimi?

1

`pul1 + pul2`, `vaqt++`, `cout << sana` — qaysi biri tabiiyroq o‘qiladi?

2

Ikki talabani `+` bilan “qo‘shish” nimani bildiradi?

3

`2 * vektor` va `vektor * 2` ikkalasi ham ishlashi kerakmi?

O‘QUV MAQSADLARI

Bu ma’ruza oxirida siz quyidagilarni bajara olasiz...

1. **Tanlash** — operatorni a’zo yoki erkin (friend) funksiya qilib yozishni
2. **Yozish** — arifmetik operatorlarni += orqali + usulida
3. **Qurish** — taqqoslash operatorlarini ikkita asosiy operator orqali
4. **Qo‘llash** — C++20 <=> = default bilan barcha taqqoslashlarni
5. **Yaratish** — oqim operatorlarini: << va xatoni tekshiruvchi >>
6. **Qayta yuklash** — [], (), ->, * va explicit bool operatorlarini
7. **Tushuntirish** — const va const bo‘lmagan operator[] zarurligini
8. **Loyihalash** — KattaSon sinfini operatorlar bilan

01

A’zo yoki erkin funksiya?

a @ b → a.operator@(b) yoki operator@(a, b) · qoidalar · simmetriya

a.operator@(b)

operator@(a, b)

friend

2 + a

QOIDALAR

Qaysi operator qayerda yoziladi?

Qoida	Operatorlar
Faqat a’zo	`=`, `[]`, `()`, `->`, tur o‘zgartirish operatorlari
Faqat erkin	Chap operand boshqa sinf: `cout << a`, `cin >> a`, `2 * a`
Odatda a’zo	Obyektni o‘zgartiruvchi: `+=`, `-=`, `++`, `--`, unar `-`
Odatda erkin (friend)	Simmetrik: `+ - * / == != < >`
Qayta yuklanmaydi	`::`, `.`, `.*`, `?:`, `sizeof`, `typeid`

Operator — maxsus nomli funksiya: kompilyator `a @ b` uchun a’zo `a.operator@(b)` va erkin `operator@(a, b)` ni qidiradi.

SIMMETRIYA

A’zo operator va 2 + a muammosi

```
struct Metr {
    double q;
    Metr(double q) : q(q) {}
    Metr operator+(const Metr &b) const { return Metr(q + b.q);
            }
};
struct Sekund {
    double q;
    Sekund(double q) : q(q) {}
    friend Sekund operator+(const Sekund &a, const Sekund &b) {
            return Sekund(a.q + b.q); }
};
int main() {
    Metr m(5);
    cout << (m + 2).q << " ";
    Sekund s(5);
    cout << (s + 2).q << " " << (2 + s).q;
}
```

▶ NATIJA

```
7 7 7
```

`m + 2`

ishlaydi: 2 → Metr(2)

`2 + m`

XATO — a’zo operatorda chap operand Metr bo‘lishi shart

`2 + s`

erkin operator — ikkala tomonda o‘zgartirish mumkin

To‘liq operatorlar to‘plami

```
class Pul {
    long long tiyin;
public:
    Pul(long long som = 0, int t = 0) : tiyin(som * 100 + t) {}
    Pul& operator+=(const Pul &b) { tiyin += b.tiyin;
            return *this; }
    Pul& operator*=(int k) { tiyin *= k; return *this; }
    friend Pul operator+(Pul a, const Pul &b) { return a += b; }
    friend Pul operator*(Pul a, int k) { return a *= k; }
    friend bool operator<(const Pul &a, const Pul &b) {
            return a.tiyin < b.tiyin; }
    friend ostream& operator<<(ostream &o, const Pul &p) {
        return o << p.tiyin / 100 << "." << p.tiyin % 100 / 10
                 << p.tiyin % 10 << " so'm";
    }
};
int main() {
    Pul non(4500), choy(2000, 50);
    Pul jami = non * 2 + choy;
    cout << jami << " " << (choy < non);
}
```

▶ NATIJA

```
11000.50 so'm 1
```

long long tiyin

aniqlik uchun — double pulda xatolik beradi

+= a’zo, + erkin

asosiy mantiq bitta joyda

non * 2 + choy

9000 + 2000.50

TUSHUNISHNI TEKSHIRISH

O‘ylang – Juftlikda – Bo‘lishing · 3 daqiqa

1-bo‘lim yakuni: A’zo yoki erkin funksiya?

1

Qaysi operatorni qayta yuklab bo‘lmaydi: +, [], ?: yoki <<?

Savol 1

2

Nima uchun `operator=` faqat a’zo bo‘ladi?

Savol 2

3

a’zo `operator+` bo‘lsa `2 + a` nima uchun ishlamaydi?

Savol 3

02

Arifmetik operatorlar

+= havola qaytaradi · + yangi obyekt · Vektor · matritsa va tez daraja

+= → +

return *this

Vektor

Matritsa

ANDOZA

Avval += (a’zo), keyin + (erkin)

operator+= — a’zo

- `Vektor& operator+=(const Vektor &b) { x += b.x; y += b.y; return *this; }`
- Obyektni o‘zgartiradi
- Havola qaytaradi — `a += b += c` zanjiri

operator+ — erkin

- `friend Vektor operator+(Vektor a, const Vektor &b) { return a += b; }`
- a — nusxa (qiymat bo‘yicha)
- Yangi obyekt qaytaradi — operandlar o‘zgarmaydi

Foyda: hisoblash mantiqi bir joyda, + va += doim bir xil natija beradi.

VEKTOR

Vektor2D arifmetikasi

```
struct Vektor {
    double x, y;
    Vektor& operator+=(const Vektor &b) { x += b.x; y += b.y;
            return *this; }
    Vektor& operator*=(double k) { x *= k; y *= k; return *this;
            }
    Vektor operator-() const { return {-x, -y}; }
    friend Vektor operator+(Vektor a, const Vektor &b) {
            return a += b; }
    friend Vektor operator*(Vektor a, double k) { return a *= k;
            }
    friend Vektor operator*(double k, Vektor a) { return a *= k;
            }
    friend double operator*(const Vektor &a, const Vektor &b) {
            return a.x * b.x + a.y * b.y; }
};
int main() {
    Vektor a{3, 4}, b{1, 2};
    Vektor c = 2 * a + -b;
    cout << c.x << "," << c.y << " " << a * b;
}
```

▶ NATIJA

```
5,6 11
```

2 * a

(6, 8) — ikkala tartib uchun alohida operator*

-b

unar minus: (-1, -2)

a * b

Vektor * Vektor — skalyar ko‘paytma: 3 + 8 = 11

MATRITSA

Fibonachchini log(n) qadamda

```
struct M {
    long long a, b, c, d;
    friend M operator*(const M &x, const M &y) {
        return {x.a * y.a + x.b * y.c, x.a * y.b + x.b * y.d,
                x.c * y.a + x.d * y.c, x.c * y.b + x.d * y.d};
    }
};
long long fib(int n) {
    M r{1, 0, 0, 1}, m{1, 1, 1, 0};
    while (n) {
        if (n & 1) r = r * m;
        m = m * m;
        n >>= 1;
    }
    return r.b;
}
int main() {
    cout << fib(10) << " " << fib(90);
}
```

▶ NATIJA

```
55 2880067194370816120
```

[[1,1],[1,0]]^n

yuqori o‘ng element — fib(n)

n & 1, n >>= 1

tez darajaga ko‘tarish (1-darsdagi bitli amallar!)

fib(90)

atigi 7 ta ko‘paytma

TUSHUNISHNI TEKSHIRISH

O‘ylang – Juftlikda – Bo‘lishing · 3 daqiqa

2-bo‘lim yakuni: Arifmetik operatorlar

1

Nima uchun operator+ ko‘pincha operator+= orqali yoziladi?

Savol 1

2

operator+= nima qaytaradi va nega?

Savol 2

3

operator+ ning birinchi parametri nima uchun qiymat bo‘yicha?

Savol 3

AMALIY TOPSHIRIQ

Har kim o‘z qurilmasida · 10 daqiqa · Oson · 10 ball

1-topshiriq: Kasr: arifmetik operatorlar

`Kasr` sinfi uchun `+`, `-`, `*`, `/` operatorlarini va `<<` ni qayta yuklang. Kasr har doim qisqartirilgan, maxraj musbat; maxraj 1 bo‘lsa faqat surat chiqadi. Nolga bo‘lishda (`/` va ikkinchi kasr 0) `XATO` chiqaring.

Kirish: Birinchi qatorda `q` ($1 \le q \le 100$), keyin `q` qatorda `a/b op c/d` (b, d ≠ 0, $|son| \le 10^4$).

Chiqish: Har bir amal natijasi.

MISOL · KIRISH

```
3
1/2 + 1/3
3/4 * 2/3
1/2 / 0/5
```

MISOL · CHIQISH

```
5/6
1/2
XATO
```

Telefon kamerasi bilan skanerlang

1. Saytga kiring (login va parol)
2. Kodni yozing va sinab ko‘ring
3. ✓ Topshirish — natija o‘qituvchiga

03

Taqqoslash operatorlari

== va < dan qolganlari · sort uchun operator< · C++20 <=>

==

<

sort

<=>

Oltita operator — ikkita asos

Operator	Ifoda
`a != b`	`!(a == b)`
`a > b`	`b < a`
`a <= b`	`!(b < a)`
`a >= b`	`!(a < b)`

operator< ayniqsa muhim: `std::sort`, `std::map`, `std::set`, `std::max` aynan shundan foydalanadi.

SANA

operator< bilan saralash

```
struct Sana {
    int k, o, y;
    friend bool operator<(const Sana &a, const Sana &b) {
        if (a.y != b.y) return a.y < b.y;
        if (a.o != b.o) return a.o < b.o;
        return a.k < b.k;
    }
};
int main() {
    vector<Sana> v = {{20, 1, 2026}, {8, 1, 2026}, {3, 2, 2026},
            {27, 12, 2025}};
    sort(v.begin(), v.end());
    for (auto &s : v) cout << s.k << "." << s.o << " ";
    cout << (max(v[0], v[1]) < v[2]);
}
```

▶ NATIJA

```
27.12 8.1 20.1 3.2 1
```

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

operator< dan foydalanadi

`27.12.2025`

yil kichik — birinchi

`max(...)`

u ham operator< ni ishlatadi

“Kosmik kema” operatori <=>

```cpp
#include <iostream>
#include <vector>
#include <algorithm>
#include <compare>
using namespace std;

struct Talaba {
    int kurs;
    double ball;
    string ism;
    auto operator<=>(const Talaba &) const = default;
};
int main() {
    vector<Talaba> t = {{2, 88.5, "Bekzod"}, {1, 91, "Aziza"},
            {2, 88.5, "Anvar"}};
    sort(t.begin(), t.end());
    for (auto &x : t) cout << x.ism << " ";
    cout << (t[0] < t[1]) << (t[1] == t[2]);
    return 0;
}
```

▶ NATIJA

```
Aziza Anvar Bekzod 10
```

= default

barcha 6 ta operator tayyor

Tartib

maydonlar e’lon tartibida: kurs → ball → ism

Anvar < Bekzod

kurs va ball teng — ism bo‘yicha

TUSHUNISHNI TEKSHIRISH

O‘ylang – Juftlikda – Bo‘lishing · 3 daqiqa

3-bo‘lim yakuni: Taqqoslash operatorlari

1

`<=>` = `default` nima beradi?

Savol 1

2

operator< qaysi standart vositalarga kerak?

Savol 2

3

`a <= b` ni faqat `<` orqali qanday yozasiz?

Savol 3

AMALIY TOPSHIRIQ

Har kim o‘z qurilmasida · 10 daqiqa · Oson · 10 ball

2-topshiriq: Sanalarni taqqoslash

`Sana` sinfi uchun `<` va `==` operatorlarini qayta yuklang (yoki C++20 `<=>`). `main()` sanalarni `sort` bilan saralaydi va takrorlanmaydigan sanalar sonini topadi — bu faqat sizning operatorlaringiz orqali ishlaydi.

Kirish: Birinchi qatorda `n` ($1 \le n \le 1000$), keyin `n` ta sana `dd.mm.yyyy` ko‘rinishida.

Chiqish: Saralangan sanalar (har biri alohida qatorda, `dd.mm.yyyy`), oxirida `Turli: k`.

MISOL · KIRISH

```
4
15.03.2025
01.12.2024
15.03.2025
31.01.2025
```

MISOL · CHIQISH

```
01.12.2024
31.01.2025
15.03.2025
15.03.2025
Turli: 3
```

Telefon kamerasi bilan skanerlang

1. Saytga kiring (login va parol)
2. Kodni yozing va sinab ko‘ring
3. ✓ Topshirish — natija o‘qituvchiga

04

Oqim operatorlari

operator<< zanjiri · tekshiruvchi operator>> · failbit · shablon <<

ostream&

istream&

setstate

template

<< VA >>

Sinfingiz cout, cin, fayl va stringstream bilan

Oqim operatorlari doim erkin funksiya bo‘ladi va oqimning o‘zini havola qilib qaytaradi — aynan shu `cout << a << b` zanjirini ta’minlaydi.

Zanjir

```
cout << a << b ≡
operator<<(operator<<(cout,
a), b)
```

Chiqarish

```
friend ostream&
operator<<(ostream &o, const
T &x)
```

O‘qish

```
friend istream&
operator>>(istream &in, T
&x)
```
— noto‘g‘ri bo‘lsa `in.setstate(ios::failbit)`

Shunda `while (cin >> v)` va `if (cin >> v)` standart turlar kabi to‘g‘ri ishlaydi.

VAQT

Tekshiruvchi operator>>

```
struct Vaqt {
    int s = 0, d = 0;
    friend istream& operator>>(istream &in, Vaqt &v) {
        int s, d; char c;
        if (in >> s >> c >> d) {
            if (c == ':' && s >= 0 && s < 24 && d >= 0
                    && d < 60) v = {s, d};
            else in.setstate(ios::failbit);
        }
        return in;
    }
};
int main() {
    for (string q : {"12:30", "25:00", "12-30", "7 : 05"}) {
        istringstream ss(q);
        Vaqt v;
        if (ss >> v) cout << v.s * 60 + v.d << " ";
        else cout << "xato ";
    }
}
```

▶ NATIJA

```
750 xato xato 425
```

25:00

soat noto‘g‘ri — failbit

12-30

ajratuvchi ':' emas — failbit

"7 : 05"

>> bo‘sh joylarni o‘tkazadi — to‘g‘ri

v o‘zgarmaydi

xato bo‘lsa obyekt buzilmaydi

Istalgan vector va pair uchun

```cpp
template <typename A, typename B>
ostream& operator<<(ostream &o, const pair<A, B> &p) {
    return o << "(" << p.first << ", " << p.second << ")";
}
template <typename T>
ostream& operator<<(ostream &o, const vector<T> &v) {
    o << "[";
    for (size_t i = 0; i < v.size(); i++) o << (i ? ", " : "")
            << v[i];
    return o << "]";
}
int main() {
    vector<pair<string, int>> v = {{"Aziza", 91}, {"Bekzod",
            78}};
    vector<vector<int>> m = {{1, 2}, {3}};
    cout << v << endl << m;
}
```

▶ NATIJA

```
[(Aziza, 91), (Bekzod, 78)]
[[1, 2], [3]]
```

o << v[i]

element turi uchun operator<< chaqiriladi — rekursiv

vector<vector<int>>

ichma-ich konteynerlar ham o‘zi ishlaydi

Shablon

bir marta yoziladi — barcha turlar uchun

TUSHUNISHNI TEKSHIRISH

O‘ylang – Juftlikda – Bo‘lishing · 3 daqiqa

4-bo‘lim yakuni: Oqim operatorlari

1

Nima uchun operator<< `ostream&` qaytaradi?

Savol 1

2

Noto‘g‘ri kiritishda operator>> nima qilishi kerak?

Savol 2

3

Oqim operatorlari nima uchun a’zo bo‘lolmaydi?

Savol 3

AMALIY TOPSHIRIQ

Har kim o‘z qurilmasida · 10 daqiqa · Oson · 10 ball

3-topshiriq: Oqim operatorlari: Nuqta

`Nuqta` uchun `>>` va `<<` operatorlarini **erkin funksiya** sifatida yozing. `>>` nuqtani `(x, y)` ko‘rinishida o‘qisin, `<<` ham shunday chiqarsin. `main()` koordinata boshidan eng uzoq nuqtani (teng bo‘lsa — birinchisini) va barcha nuqtalar yig‘indisini chiqaradi.

Kirish: Birinchi qatorda `n` ($1 \le n \le 1000$), keyin `n` ta nuqta `(x, y)` ko‘rinishida (butun sonlar, $|son| \le 10^6$).

Chiqish: Ikki qator: eng uzoq nuqta va yig‘indi nuqta — ikkalasi `(x, y)` ko‘rinishida.

MISOL · KIRISH

```
3
(1, 2)
(-3, 4)
(0, 0)
```

MISOL · CHIQISH

```
(-3, 4)
(-2, 6)
```

Telefon kamerasi bilan skanerlang

1. Saytga kiring (login va parol)
2. Kodni yozing va sinab ko‘ring
3. ✓ Topshirish — natija o‘qituvchiga

05

Maxsus operatorlar

[] ikki versiya · () funktor/matritsa · -> va * · explicit bool

operator[]

operator()

operator->

explicit bool

MAXSUS

Konteyner va ko‘rsatkichga o‘xshash sinflar

Operator	Imzo	Qo‘llanishi
`[]`	`T& operator[](size_t) + const T& ... const`	`a[i] = 5;` va const obyektdan o‘qish
`()`	`R operator()(args...)`	Funktor, `m(i, j)` — istalgan sondagi parametr
`->, *`	`T* operator->()`, `T& operator*()`	Aqlli ko‘rsatkichlar, iteratorlar
`bool`	`explicit operator bool() const`	`if (obj)` — “yaroqlimi?”

explicit bo‘lmasa `int n = obj;`, `obj + 1` kabi ma’nosiz ifodalar ham kompilyatsiya bo‘ladi (bool → int).

XAVFSIZ MmassiV

[] ikki versiya va explicit bool

```
class XMassiv {
    vector<int> d;
public:
    XMassiv(initializer_list<int> q) : d(q) {}
    int& operator[](size_t i) {
        if (i >= d.size()) throw out_of_range("indeks");
        return d[i];
    }
    const int& operator[](size_t i) const { return d.at(i); }
    explicit operator bool() const { return !d.empty(); }
};
int main() {
    XMassiv a{10, 20, 30};
    a[1] = 99;
    const XMassiv &c = a;
    cout << c[1] << " " << (a ? "bor" : "yo'q") << " ";
    try { a[5] = 1; } catch (const out_of_range &) { cout
            << "chegara!"; }
}
```

▶ NATIJA

```
99 bor chegara!
```

int& operator[]

havola — `a[1] = 99` yozish mumkin

const versiya

const obyektdan faqat o‘qish: `c[1] = 5` — XATO

explicit bool

`if (a)` / `a ? ...` ishlaydi, `int n = a;` — yo‘q

-> VA *

O‘z aqlli ko‘rsatkichimiz

```
struct Talaba {
    string ism;
    int ball;
};
template <typename T>
class Egasi {
    T *p;
public:
    explicit Egasi(T *q) : p(q) {}
    ~Egasi() { delete p; cout << " [o'chirildi]"; }
    Egasi(const Egasi &) = delete;
    T* operator->() const { return p; }
    T& operator*() const { return *p; }
};
int main() {
    Egasi<Talaba> t(new Talaba{"Aziza", 90});
    t->ball += 5;
    cout << (*t).ism << ": " << t->ball;
}
```

▶ NATIJA

```
Aziza: 95 [o'chirildi]
```

t->ball

operator-> ko‘rsatkichni qaytaradi, keyin ->ball

(*t).ism

operator* havolani qaytaradi

~Egasi()

RAII — delete avtomatik (soddalashtirilgan unique_ptr)

TUSHUNISHNI TEKSHIRISH

O‘ylang – Juftlikda – Bo‘lishing · 3 daqiqa

5-bo‘lim yakuni: Maxsus operatorlar

1

Nima uchun operator[] ning const va const bo‘lmagan versiyasi yoziladi?

Savol 1

2

operator bool nima uchun explicit bo‘lishi kerak?

Savol 2

3

operator-> nima qaytaradi?

Savol 3

AMALIY TOPSHIRIQ

Har kim o‘z qurilmasida · 10 daqiqa · O‘rta · 20 ball

4-topshiriq: Ko‘phad: [] va ()

`Kophad` sinfi koeffitsiyentlarni saqlaydi. `operator[]` `i`-darajali koeffitsiyentga murojaat qilsin (o‘zgartirish ham mumkin), `operator()` esa ko‘phadning `x` nuqtadagi qiymatini **Gorner sxemasi** bilan hisoblasin.

Kirish: Birinchi qatorda daraja `n` ($0 \le n \le 20$) va `n + 1` ta koeffitsiyent (`a0 a1 … an`). Keyin `q` va `q` ta buyruq: `? x` — qiymatni hisoblash, `= i c` — `i`-koeffitsiyentni `c` ga o‘zgartirish. Natijalar `long long` ga sig‘adi.

Chiqish: Har bir `?` uchun ko‘phad qiymati.

MISOL · KIRISH

```
2 1 0 1
3
? 2
= 0 -4
? 2
```

MISOL · CHIQISH

```
5
0
```

Telefon kamerasi bilan skanerlang

1. Saytga kiring (login va parol)
2. Kodni yozing va sinab ko‘ring
3. ✓ Topshirish — natija o‘qituvchiga

06

Loyiha: KattaSon

Cheksiz aniqlikdagi butun son · ustunli qo‘shish va ko‘paytirish · oltin qoida

vector<int> | Eltish | n! | Oltin qoida

G‘OYA

Raqamlarni massivda saqlaymiz

1

Muammo

long long ≈ 9.2×10^{18} gacha; 25! yoki 2^{100} sig‘maydi

2

Saqlash

`vector<int> r` — raqamlar teskari: r[0] — birlar

3

Qo‘shish

Maktabdagi “ustun bilan”: raqam + raqam + eltish

4

Ko‘paytirish

Har raqam juftligi: `c[i + j] += a[i] * b[j]`

5

Operatorlar

`a + b`, `a * b`, `cout << a` — int kabi

Oltin qoida: operator kutilgan ma’noda ishlasin. Ma’nosi noaniq bo‘lsa (ikki talabani “qo‘shish”) — oddiy nomli metod yozing.

QO‘SHISH

KattaSon::operator+=

```
class KattaSon {
    vector<int> r;
public:
    explicit KattaSon(const string &s) {
        for (int i = s.size() - 1; i >= 0;
                i--) r.push_back(s[i] - '0');
    }
    KattaSon& operator+=(const KattaSon &b) {
        int elt = 0;
        for (size_t i = 0; i < max(r.size(), b.r.size()) || elt;
                i++) {
            if (i == r.size()) r.push_back(0);
            int s = r[i] + elt + (i < b.r.size() ? b.r[i] : 0);
            r[i] = s % 10; elt = s / 10;
        }
        return *this;
    }
    friend KattaSon operator+(KattaSon a, const KattaSon &b) {
            return a += b; }
    friend ostream& operator<<(ostream &o, const KattaSon &k) {
        for (int i = k.r.size() - 1; i >= 0; i--) o << k.r[i];
        return o;
    }
};
int main() {
    KattaSon a("98765432109876543210"),
            b("12345678901234567890");
    cout << a + b;
}
```

▶ NATIJA

```
111111111011111111100
```

`s % 10, s / 10`

raqam va eltish

`|| elt`

oxirgi eltish uchun yangi raqam

`+= → +`

yana o‘sha andoza

30! — 33 xonali son

```
struct KS {
    vector<int> r{1};
    KS& operator*=(int k) {
        int elt = 0;
        for (size_t i = 0; i < r.size() || elt; i++) {
            if (i == r.size()) r.push_back(0);
            long long s = (long long)r[i] * k + elt;
            r[i] = s % 10; elt = s / 10;
        }
        return *this;
    }
    friend ostream& operator<<(ostream &o, const KS &k) {
        for (int i = k.r.size() - 1; i >= 0; i--) o << k.r[i];
        return o;
    }
};
int main() {
    KS f;
    for (int i = 2; i <= 30; i++) f *= i;
    cout << f;
}
```

▶ NATIJA

```
265252859812191058636308480000000
```

r{1}

boshlang‘ich qiymat 1

f *= i

har qadamda kichik songa ko‘paytirish

30!

≈ 2.65×10^{32} — long long dan 10^{14} marta katta

TUSHUNISHNI TEKSHIRISH

O‘ylang – Juftlikda – Bo‘lishing · 3 daqiqa

6-bo‘lim yakuni: Loyiha: KattaSon

1

Operatorlarni qayta yuklashning oltin qoidasi?

Savol 1

2

KattaSon raqamlarni nima uchun teskari tartibda saqlaydi?

Savol 2

3

Qo‘shishda “eltish” nima?

Savol 3

XULOSA

Asosiy xulosalar

1 A’zo yoki erkin

`= [] () ->` — a’zo; `<< >>` va simmetrik — erkin (friend).

2 += orqali +

`+=` havola qaytaradi, `+` — nusxa + `+=` va yangi obyekt.

3 Taqqoslash

`==` va `<` yetarli; C++20 `<=>` = `default` — hammasi.

4 Oqimlar

`ostream&/istream&` qaytaradi; xato — `setstate(failbit)`.

5 Maxsus

`[]` — ikki versiya; `explicit operator bool`; `->/*` — aqlli ko‘rsatkich.

6 Oltin qoida

Operator kutilgan ma’noda ishlasin — aks holda nomli metod.

TAKRORLASH

Yakunlashdan oldin sakkizta savol

1. `cout << a` uchun operator qanday e’lon qilinadi?
2. operator+= nima qaytaradi?
3. operator+ nima qaytaradi?
4. `a > b` ni `<` orqali?
5. `<=> = default` qaysi tartibda taqqoslaydi?
6. operator[] ning const versiyasi nima qaytaradi?
7. explicit operator bool qayerda ishlaydi?
8. Qayta yuklanmaydigan 2 operator?

LABORATORIYA ISHI

14-laboratoriya ishi: 6 ta dasturlash topshirig‘i

5 **Matritsa operatorlari**
O‘rta · 20 ball

6 **Vaqt: ++ va -**
O‘rta · 20 ball

7 **Funktor: filtr**
O‘rta · 20 ball

8 **KattaSon: qo‘shish va taqqoslash**
O‘rta · 20 ball

9 **KattaSon: ko‘paytirish va faktorial**
Qiyin · 30 ball

10 **Versiyalar: <=> operatori**
O‘rta · 20 ball

Telefon kamerasi bilan skanerlang

Kod saytda yoziladi va server tomonidan avtomatik tekshiriladi.

O‘qituvchi kodingizni ko‘rib, baho qo‘yadi.

CHIQISH CHIPTASI

Darsni tark etishdan oldin

Javoblaringizni qog‘ozga yozing va o‘qituvchiga topshiring.

1

`Kasr` uchun qaysi operatorlarni a’zo, qaysilarini erkin qilasiz?

2

Nima uchun operator>> xato kiritishda obyektni o‘zgartirmasligi kerak?

3

O‘zingiz yaratmoqchi bo‘lgan sinf va unga mos operatorni ayting.