**Условие**

Даден е клас `Vector4D`, представляващ наредена четворка с реални числа. Добавете декларации и дефиниции на необходимите оператори, така че да бъдат възможни следните операции по следните правила:

* Достъп

Vector4D v = Vector4D(a, b, c, d);

v[0] == a;

v[1] == b;

v[2] == c;

v[3] == d;

// Мутирането също трябва да е възможно:

v[0] = x; // вече v е наредената четворка (x, b, c, d)

* Събиране
  + Vector4D(a, b, c, d) + Vector4D(i, j, k, l) == Vector4D(a+i, b+j, c+k, d+l)
  + Аналогично за +=
* Изваждане
  + Vector4D(a, b, c, d) - Vector4D(i, j, k, l) == Vector4D(a-i, b-j, c-k, d-l)
  + Аналогично за -=
* Умножение
  + Поелементно умножение: Vector4D(a, b, c, d) \* Vector4D(i, j, k, l) == Vector4D(a\*i, b\*j, c\*k, d\*l)
  + Умножение със скалар (double) **отдясно**: Vector4D(a, b, c, d) \* x == Vector4D(x\*a, x\*b, x\*c, x\*d)
  + Аналогично за \*=
* Деление
  + Vector4D(a, b, c, d) / Vector4D(i, j, k, l) == Vector4D(a/i, b/j, c/k, d/l)
  + Аналогично за /=
* Сравнение
  + == и !=: За да са равни две наредени четворки, трябва да съвпадат поелементно
  + <, >, <=, >=: Сравняват елементите лексикографски
  + Освен това нека и !v == true тогава и само тогава, когато v == Vector4D(0, 0, 0, 0)
* Отрицание
  + -Vector4D(1, 2, 3, 4) == Vector4D(-1, -2, -3, -4)