```
class Time {
private:
    unsigned short _hours;
    unsigned short _minutes;
    unsigned short _seconds;

public:
          // Operator Overloading
          // (++, --) (Postfix, Prefix)
          // (<, >, <=, >=, !=, ==)

          // () => print evezi ishlesin.

          // hamisi 0-a beraber olsa => true
          bool operator!() const;
};



class Vector {
private:
    int* _array = nullptr;
    size_t _size = 0;
    size_t _capacity = 15;

public:
    Vector() = default;
    Vector(size_t capacity);

    int* getData() const { return _array; }
    size_t size() const { return _size; }
    size_t capacity() const { return _capacity; }

    void print() const;

    int& operator[](size_t index);
    int& operator()(size_t index);
    Vector& push_back(const int value);
    Vector& push_front(const int value);

    Vector& pop_back();
    Vector& pop_front();

    void delete_by_index(const size_t index);
    void insert_by_index(const size_t index, const int element);

    // tapsa index-ni qaytarsin,
    // tapmasa -1 (unsigned int-in max qiymeti qayidacaq)
    size_t find(const int element);
    size_t rfind(const int element);

    // reverse true gonderilse, tersine sort olunsun.
    void sort(bool reverse = false);
```

Operator overloading

1) operator+ 2 Vector-daki array-leri birleshdirib, yeni Vector qaytarmaq

[1,2] + [3,4,5] = [1,2,3,4,5]}

2) operator== 2 Vector-daki array-larin sayi ve ededler beraberdi ya yox?

[1,4,2] == [1,4,2] => true

3) operator!= yuxarda yazilanin eksi

4) operator++ {array-daki butun ededleri 1 vahid artirmaq} {Postfix, Prefix}

5) operator-- {array-daki butun ededleri 1 vahid azaltmaq} {Postfix, Prefix}

NOTE: Yazilan kodunun uzerinə, qeyd olunanlari elavə etmək lazimdi.

```
};
```

NOTE

1) Bütün yoxlamalari etmək.

2) Lazim olan yerlerde keçdiymiz dersleri istifade etmek. (Mütləq)

UGURLAR 😊😊😊