

Реализирайте функция (с подаден `std::vector`) или метод на клас “динамичен масив”, която/който:

1. По подаден елемент **elem** връща итератор към първия намерен елемент от масива равен на **elem** или итератор към края на масива ако не съдържа **elem**.

2. Връща копие на масива, но с елементите му в обратен ред.

Вход: `[ 1, 5, 6, 2, 0 ]`

Изход: `[ 0, 2, 6, 5, 1 ]` *// нов масив*

3. По подаден масив проверява дали всички елементи от втория масив се съдържат в първия.

Вход: `( [ 1, 5, 6, 2, 0 ], [ 0, 5, 1 ] )`

Изход: `true`

4. По подадена функция прилага функцията върху всеки елемент на масива.

Пример: `void square( int& num ) { num *= num; }`

Вход: `( [ 1, 5, 6, 2, 0 ], square )`

Изход: `[ 1, 25, 36, 4, 0 ]`

5. По подаден предикат (булева функция) връща нов масив съдържащ тези елементите на оригиналния масив, които удовлетворяват предиката.

Пример: `bool isEven( int num ) { return num % 2 == 0; }`

Вход: `( [ 1, 5, 6, 2, 0 ], isEven )`

Изход: `[ 6, 2, 0 ]` *// нов масив*

6. По подадено число  $k \in [1; n]$  връща итератор към **k**-тия по големина елемент от масива.

Сложността по време да бъде  $O(n * k)$ .

7. Връща нов масив съдържащ само неповтарящите се елементи на оригиналния.

Сложността по време да бъде  $O(n * \log(n))$ .