

Homework

24.1. (easy)

Реализовать шаблонную функцию принимающую на вход массив одного типа вместе с его размером и выводящую динамически выделенный массив второго типа, содержащий три самых больших элемента, приведенных к другому типу. Попробовать передавать статический и динамические массивы в качестве аргумента.

Произвести тесты такого типа

(*тип входного массива* -> *тип массива на выходе*)

Int -> int

Double -> double

Double -> int

Char -> char

Char -> char

Объявление шаблонной функции:

```
template<class T1, class T2>
T2 * searchThreeMax(const T1 * array);
```

Вызов: (входной массив может быть динамическим, может быть разного размера, может быть неинициализирован вовсе. Обработайте все варианты (везде где это возможно, сделайте проверку на дурака) . В случае невозможности набрать 3 максимальных элемента выходной массив должен содержать 3 нуля ({0, 0, 0})).

```
double arrD[5]={123.542,75.5,98.875,5321.2,5.123};
int * arrI = searchThreeMax<double,int>(arrD,5);
```

В вышеуказанном случае входной массив типа double, результирующий – int. То есть кейс double -> int. Результат работы функции в вышеуказанном случае будет **динамически выделенный!!!!** массив {5321, 123, 98}.

24.2 (medium)

Реализовать класс ArrayWrapper

Полями данных которого будет double * m_array, размер массива uint64_t m_size, который будет в эту переменную выделен динамически и std::string m_name переменная, которая будет содержать имя массива.

Перегрузить оператор сравнения для этого класса (подумайте, по какому принципу может производиться сравнение между объектами этого класса. **Не по размеру массива!!**) , подумать, по какому дополнительному параметру могут сравниваться два объекта если они оказались равны по первому.

Перегрузить конструктор по умолчанию, конструктор копирования, конструктор перемещения(перемещения если не делали, то можно не делать, объясню на занятии). Перегрузить оператор копирования присваиванием, оператор перемещения присваиванием(можно не делать если не знаете), подумайте, на что должна быть проверка, при выполнении этих операторов. Реализовать дополнительный конструктор, позволяющий инициализацию объекта при помощи ввода через консоль. Перегрузить оператор вывода в консоль << и, естественно, деструктор.

```
class ArrayWrapper{
private:
    double * m_array;
    uint64_t m_size;
    std::string m_name;
};
```

24.3 (medium)

Далее - динамически выделить массив из объектов класса ArrayWrapper и применить к нему функцию реализованную в первом задании. Таким образом :

ArrayWrapper-> ArrayWrapper

ArrayWrapper-> double (hard. Подумайте, какой дополнительный оператор для этого нужно перегрузить)

24.4 (easy) optionally

Изучить что такое алгоритмическая сложность и предоставить оценку для созданной функции по количеству операций

Гуглить исключительно на английском языке.