**Динамические массивы**

В следующих двух задачах требуется использовать размещение массивов в динамической памяти (new / delete).

* Создать функцию генерации динамического массива заданной длины. Для заполнения значений использовать генератор случайных чисел std::rand().
* \* Создать массив из 1 000 000 целых чисел. Замерить время сортировки массива функцией.
* Реализовать функцию вставки значения в середину динамического массива. Выделять память под массив нового размера. Если число элементов в массиве нечётное, вставить элемент после центрального.
* Реализовать функцию удаления дубликатов из массива. Освобождать неиспользуемую память.
* Дан массив целых чисел *A*, создать массив, в который входят только чётные элементы *A*. Вернуть новый массив и его длину.  Поскольку под возвратом массива понимается возврат указателя, то int \* можно сделать либо явным возвращаемым значением функции, либо передавать по ссылке, подразумевая выходной параметр. Следовательно, имеем два приемлемых варианта заголовка данной функции:

int \* onlyEvens(int \* a, int size, int & newSize);

void onlyEvens(int \* a, int size, int \* & evens, int & newSize);

***Указание (алгоритм).****Есть два подхода к решению подобных задач. Либо предварительно подсчитать количество чётных элементов и выделить нужное количество памяти под результирующий массив, либо выделить под результирующий массив максимально возможное количество памяти, которое может потребоваться (в данном случае это длина исходного массива). Имеем традиционный выбор между затратами по памяти в обмен затрат по времени или наоборот. Оба варианта приемлемы.*

* *Дан массив целых чисел A, создать массив, в котором после каждого элемента, последняя цифра которого равна трём, вставлено число 0.*
* *Дан массив вещественных чисел. Удалить из массива все элементы, меньшие среднего арифметического элементов массива.*