Контрольная работа. Массивы и сортировка. 8Д.

В каждой из предложенных ниже задач, нужно реализовать те функции, которые описаны в условии. Функции должны называться, иметь точно такие же аргументы и возвращать те же значения, как написано в условии. В противном случае, система проверки не зачтет ваше решение.
Прототипы функций, которые нужно реализовать, писать **НЕ** надо. Система сама вставит их.
Также **НЕ** надо писать `main()`.
Если для реализации нужных функций вам понадобятся вспомогательные, используйте их. Для них заголовки писать можно и нужно.
К каждому решению система автоматически подключит библиотеки `stdio.h` и `iostream`. Если вам понадобятся другие библиотеки, подключайте их самостоятельно. Не стоит подключать `TXLib.h`, система вас не поймет.
Запрещается использовать стандартные функции сортировки, реализованные не вами.

A. *Второй максимум.* Реализуйте функцию
`int SecondMax(int array[], int size),`
которая находит второе по величине число массива. Она должна возвращать значение (не индекс). Если в массиве несколько максимальных элементов, то функция должна вернуть максимум (второй по величине элемент совпадает с максимальным).

B. *Переставьте соседей.* Реализуйте функцию
`void SwapNeighbors(int array[], int size),`
которая меняет местами соседние элементы массива. 0-ой с 1-м, 2-й с 3-м .... Если в массиве нечетное количество элементов, последний должен остаться без изменений.
Также необходимо реализовать функцию
`void PrintArray(int array[], int size),`
которая печатает массив `array` на одной строчке через пробел.

C. *Является ли подпоследовательностью.* Последовательность y_i называется *подпоследовательностью* последовательности x_i, если существует возрастающая последовательность индексов $j_1 < \ldots < j_k$, такие что $y_i = x_{j_i}, \forall i$. Например, последовательность $\{1, 3\}$ является подпоследовательностью последовательности $\{1, 2, 3\}$. А последовательность $\{3, 2\}$ — нет.
Реализуйте функцию
`bool IsSubsequence(int Array[], int Size, int Sub[], int SubSize),`
которая возвращает `true`, если `Sub` является подпоследовательность `Array`, и `false` в противном случае.

D. *Сортировка в обратном порядке.* Реализуйте функцию, которая сортирует массив с целыми числами в порядке убывания.
`void ReverseSort(int array[], int size);`
Также нужно реализовать функцию
`void PrintArray(int array[], int size),`
которая печатает массив `array` на одной строчке через пробел.

E. *Последовательности из одинаковых элементов.* Необходимо узнать, состоят ли два массива из одних и тех же элементов, или в одном из них есть элемент, который не встречается в другом.
Вы должны реализовать две функции
`void MySort(int array[], int size)`, и
`bool HaveSameElements(int array1[], int size1, int array2[], int size2).`
Функция `MySort()` должна быть любой известной вам сортировкой.
Функция `HaveSameElements()` должна проверять для отсортированных массивов `array1` и `array2` состоят ли они из одних и тех же элементов. При этом каждый массив должен проходиться только один раз.