

Слияние массивов.

Домашнее задание:

1. Зарегистрироваться на [сайте](http://informatics.mccme.ru/) (<http://informatics.mccme.ru/>) и прислать мне по почте или сообщить в следующий раз на занятии логин, указанный при регистрации. В дальнейшем я добавлю вас в группу и мы будем решать выложенные там задачи.

2. Написать слияние массивов. Массив называется *отсортированным*, если его элементы упорядочены. В частности, массив отсортирован по возрастанию, если каждый следующий элемент строго больше предыдущего (то есть для любого $i > 1$ выполняется $\text{arr}[i] > \text{arr}[i - 1]$). Однако если в массиве есть одинаковые элементы, то его нельзя отсортировать по возрастанию, поэтому чаще используют массивы отсортированные по неубыванию - то есть массивы, в которых каждый следующий элемент не меньше предыдущего (то есть для любого $i > 1$ выполняется $\text{arr}[i] \geq \text{arr}[i - 1]$). При отсутствии равных элементов это одно и то же.

Задание состоит в том, чтобы объединить два отсортированных по неубыванию массива так, чтобы получившийся массив тоже был отсортирован. Пусть например первый массив $\mathbf{a1} = (10, 20, 30, 40, 50)$, второй массив $\mathbf{a2} = (15, 35, 55, 65)$, тогда в результате слияния должен получиться массив $\mathbf{a_sum} = (10, 15, 20, 30, 35, 40, 50, 55, 65)$.

P.S. Первый пункт этого домашнего задания важнее второго и обязателен к выполнению.

Краткая инструкция по регистрации: в поле «Логин» впишите логин, состоящий только из латинских букв и цифр (например свою фамилию и имя латиницей), в поле «Пароль» – любой пароль, который не забудете. Укажите личные данные (почту не обязательно), класс и школу. Класс – 6, а школу можете указать либо свою, либо ФТШ. Затем нажмите кнопку «Сохранить». В случае, если выбранный вами логин уже занят другим пользователем – используйте другой (например измените выбранный добавлением произвольного числа в конец).

P.P.S. Ниже представлен алгоритм решения задачи слияния массивов, уже рассказанный на занятии, а потому написанный достаточно сжато.

Программа должна считывать массивы из консоли (они вводятся отсортированными), после чего делать слияние и выводить результат.

Для чтения какого-либо массива \mathbf{arr} нужно сначала командой `readln(n)`; прочитать количество его элементов, после чего в цикле (`for i:=1 to n do...`) считывать его элементы командой `readln(arr[i]);`.

Для вывода массива в аналогичном цикле нужно печатать его элементы командой `write(arr[i], ' ');`. Она выводит в консоль i -ый элемент массива и пробел после него.

Алгоритм слияния основан на том, чтобы заполнять результирующий массив $\mathbf{a_sum}$ слева направо по одному элементу. Тогда на каждом шаге добавляемый элемент - минимальный из тех, которые ещё не добавлены в массив (так как этот массив должен получиться отсортированным). А минимальный неиспользованный элемент из всех, находящихся в данных массивах $\mathbf{a1}$ и $\mathbf{a2}$ – это самый левый неиспользованный в одном из них (так как они отсортированы). Таким образом, чтобы определить добавляемый в массив $\mathbf{a_sum}$ элемент достаточно сравнить самые левые неиспользованные элементы данных массивов и выбрать меньший из них. Кроме этого нужно предусмотреть, что в какой-то момент все элементы одного из данных массивов могут оказаться добавлены в итоговый. В этом случае нужно не сравнивать элементы, а добавлять их из оставшегося массива (либо, как я показывал на занятии, учесть это в условии сравнения элементов).