Лабораторная работа № 3***Генерация перестановок***

Дано конечное множество A. Требуется сгенерировать все возможные перестановки его элементов в лексикографическом порядке (*по материалам главы 1, п. 1.3.6, и главы 2, п. 2.2.1*).Требования к заданию множества – в нем не должно быть повторяющихся элементов, кроме того, удобнее использовать или только буквы, или только цифры.

Программа должна сначала упорядочить все элементы заданного множества по возрастанию (это первый – минимальный – набор), затем – посредством МИНИМАЛЬНО ВОЗМОЖНЫХ ПЕРЕСТАНОВОК! – сгенерировать последовательно возрастающие (лексикографически) наборы, вплоть до последнего, в котором все элементы упорядочены по убыванию.

Следует оценивать количество возможных перестановок и в случае, если они не поместятся на экран, выполнять их вывод в файл с выдачей на экран соответствующей информации для пользователя и выполнять поэкранный вывод с ожиданием нажатия клавиши.

Дополнительно: Предоставить пользователю возможность выбора другого варианта работы программы, в котором за исходную точку упорядочивания наборов выбирается не минимальный набор, а набор в таком порядке, как он задан пользователем.

Возможный алгоритм решения (Пример: множество А={1, 2, 3, 4, 5, 6}, |A| = n):

Предположим, что уже построено m наборов. Тогда для получения m+1-го набора:

1. Выполняется проверка последнего (m-го) набора на наличие в его конце некоторого количества символов, упорядоченных по убыванию – пусть это символы ak+1…an.   
    3 5 2 **6 4 1**– k=3, символы с 4-го по 6-й упорядочены по убыванию.
2. Если такое k найдено, то поменять местами k-й элемент и наименьший элемент из ak+1…an, больший этого ak.   
   В нашем примере это 2 и 4:  3 5 4 6 2 1 (это промежуточный набор).
3. После шага 2 упорядочить элементы с k+1-го до последнего по возрастанию. Получен очередной набор  выдать его на печать.   
    3 5 4 1 2 6.
4. Если на шаге 1 ответ отрицательный, то поменять местами 2 последних элемента и выдать на печать полученный набор. В частности, после шага 3 это неизбежное действие, т.к. все последние элементы были размещены по возрастанию  целесообразно после выполнения ш.3 задавать признак его выполнения, который будет анализироваться (и сбрасываться) на шаге 1. После шага 3 было  3 5 4 1 **2 6** выдать  3 5 4 1 6 2 .   
   Если был набор  3 5 2 6 **1 4** выдать  3 5 2 6 4 1 .
5. Если полученный набор не последний (упорядоченный по убыванию), то возврат на шаг 1. В противном случае конец работы.