

Блок 3. Машины (обязательные) по теме «Векторы»

Требования к сдаче программ, начиная с 3-го блока:

- 1) Во всех задачах верхние границы массивов объявлять **константами с указанными ниже** (в условиях задач) **значениями**, иначе проверять не будем;
- 2) В 1-ой строке исходного файла с расширением PAS разместить следующую директиву компилятору (внутри фигурных скобок пробелов быть не должно): **{\$R+,B+}**
Именно так должна выглядеть 1-ая строка текста любой вашей программы – это позволит выполнить *компиляцию в режиме контроля диапазонов и полного вычисления логических выражений*, как это требуется в Стандарте Паскаля).

8.51 (n=10)

8.53 (n=10) Организовать двойной цикл для перебора всевозможных пар точек; *требование*: каждая пара должна быть рассмотрена только 1 раз; т.е. нужно использовать цикл вида `for i:=1 to n-1 do for j:=i+1 to n do ...`

8.54 (n=10) За начальное значение для минимума взять MaxInt; для каждой последовательности завести свой массив; во внешнем цикле с заголовком `for i:=1 to n do` перебираем все элементы первого массива, и для каждого такого элемента запускаем вспомогательный цикл по поиску совпадающего элемента во втором массиве; *требование*: при решении задачи не делать лишних сравнений, т.е. досрочно прерывать внутренний цикл при первом же совпадении

8.55 (n=10) Нужен двойной цикл для перебора всевозможных пар чисел, т.е. цикл вида `for i:=1 to n-1 do for j:=i+1 to n do ...` во внешнем цикле - очередной левый элемент из пары, а во внутреннем – всевозможные элементы справа от него

8.56 Завести вспомогательный массив с индексами от 'a' до 'z' для хранения числа вхождений каждой латинской буквы

8.57 (n=10) Т.е. нужно печатать только первые вхождения каждого символа; внимание, запрещено решать задачу исходя из знаний размера используемой кодовой таблицы

8.58 (n=10) Запрещено решать задачу исходя из знаний размера используемой кодовой таблицы, т.е. надо решать по мотивам семинарской задачи **8.35**

8.29ж (n=10) Программа вводит элементы исходного массива, преобразует этот массив, и распечатывает его новое содержимое. В решении запрещено использовать вспомогательный массив; требуемое преобразование следует сделать за один просмотр массива. Внимание: в результате преобразований ненулевые элементы должны располагаться в прежнем порядке их вхождения в массив

8.41 б (n=10) Обратить внимание на замечание (в скобках) к этому пункту!

8.41 в (n=10)

8.41 г (n=10) (перед написанием программы следует разобрать задачи **8.36, 8.38, 8.40**)

| |
|--|
| Итого: 11 обязательных задач. Срок сдачи: до 5 ноября включительно. |
|--|

Блок 3. Машины (дополнительные) по теме «Векторы»

8.59 (5 очков)

Примеры работы программы:

1) sas, aaas, saaa, sb, bbbbbb, b. → s

2) qw, azz, qwwwq, wer, ert, edf, ok, aaaaa. → e w

Подсказка для этой задачи - завести два вспомогательных массива со следующим описанием

var

L: array['a'..'z'] of integer;

{накапливает число вхождений букв в словах последовательности}

W: array['a'..'z'] of Boolean;

{рабочий массив, хранит для каждого слова информацию о буквах: какие из них входят в слово (этот массив нужен, чтобы повторяющуюся в некотором слове букву не учитывать несколько раз в массиве L)}

8.39 (5 очков) (размерность массива оформить как константу **n=10**) Работа программы сводится к запросу числа (от **1** до **11**), которого не будет в массиве, после чего программа генерирует массив нужного вида и распечатывает (построчно) его элементы. Далее программа, в предположении, что отсутствующий элемент неизвестен, находит его методом бинарного поиска и выводит найденный ответ на экран. Оба числа при правильном решении должны совпасть.