**Информатика 10 класс**

### Конспект урока "Типовые задачи обработки массивов: заполнение массивов, поиск максимального и минимального элемента" (закрепление материала)

### (на основе данного текста составьте краткий конспект на 1-1,5 стр максимум, разделив данный текст на таблицу )

|  |  |
| --- | --- |
| Код программы | Пояснение строк |
|  |  |

Ссылка на дополнительные материалы <https://youtu.be/lmeu8FaIC84>

**Вопросы:**

·     Заполнение массивов.

·     Поиск максимального и минимального элементов в массиве.

Часто встречаются задачи, в которых необходимо каким-то образом заполнить двумерный массив. Рассмотрим одну из них.

**Задача:** Заполнить квадратную матрицу из символов размерностью 10×10 элементов таким образом, чтобы выше главной диагонали располагались символы **x**, на главной диагонали – символы **y**, а ниже главной диагонали – символы **z**. Вывести заполненную матрицу на экран.

Главной диагональю квадратной матрицы называется линия, которую можно провести от первого элемента первой строки к последнему элементу последней строки. Таким образом, на главной диагонали располагаются элементы матрицы, выделенные красным цветом. Выше главной диагонали располагаются элементы синего цвета, а ниже главной диагонали – элементы зелёного цвета. При решении задачи важно понимать, что в индексах элементов массива, находящихся на главной диагонали, номер строки, назовём его **i**, равен номеру столбца, назовём его **j**. В элементах, находящихся выше главной диагонали, **i** **< j**, в элементах находящихся ниже главной диагонали – напротив **i** **< j**.

Напишем программу для решения задачи. Назовём её **matrica**. Для работы программы объявим двумерный массив **a**, размерностью 10 на 10 элементов символьного типа **char**. Так же, нам понадобятся переменные **i** и **j**, в которых будем перебирать индексы элементов массива. Они будут принимать значения на промежутке от 1 до 10, поэтому объявим их, принадлежащими к целочисленному типу **byte**.

Запишем логические скобки. Тело программы будет начинаться с цикла **for i:=1 to 10 do**, который будет перебирать номера строк массива. В нём будет вложенный цикл **for j:=1 to 10 do**, который будет перебирать номера столбцов массива. Внутри цикла запишем условный оператор, который будет проверять, находится ли элемент массива на главной диагонали. Его условием будет **i = j**. После слова **then** в этом условном операторе будет следовать оператор присваивания элементу массива с индексом **i**, **j** символа **y**. После слова **else** будет следовать ещё один условный оператор, который будет проверять находится ли элемент массива выше главной диагонали. Его условием будет **i < j**. После слова **then** в нём будет следовать оператор присваивания элементу **a[i, j]** символа **x**. Если условие в этом операторе не будет выполняться, значит элемент находится ниже главной диагонали, поэтому после слова **else** запишем оператор присваивания текущему элементу массива символа **z**.

Теперь запишем цикл для вывода элементов массива. Это будет цикл **for i:=1 to 10 do**. В этом цикле в логических скобках будет записан такой же цикл, но с параметром **j**. Он будет содержать оператор **write**, который будет выводить на экран элемент **a[i, j]**. После внутреннего цикла будет следовать оператор **writeln** без параметров, для перехода на следующую строку.

**program**matrica;

**var**

a: **array**[1..10, 1..10] **of**char;

 i, j: byte;

**begin**

**for**i:=1 **to**10 **do**

**for**j:=1 **to**10 **do**

**if**i=j

**then**a[i, j]:='y'

**else if**i<j

**then**a[i, j]:='x'

**else**a[i, j]:='z';

**for**i:=1 **to**10 **do**

**begin**

**for**j:=1 **to**10 **do**

write (a[i, j]);

  writeln ();

**end**;

**end**.

**Домашняя работа: написать в тетради программу для данной задачи, написав пояснения что выполняется в каждой строке программы**

**Задача:** В заданном одномерном массиве из **n** целых чисел найти элемент с наибольшим значением и вывести его номер на экран. Если таких элементов несколько – вывести номер первого из них.

Для решения этой задачи мы будем проверять, элементы массива от первого к последнему.