**Консультация для учителей района**

**по теме: «Работа со строками при решении задач ЕГЭ»**

**учителя информатики и ИКТ МБОУ ФМЛ Барулиной Н.Н.**

 **Строка в Паскале** – упорядоченная последовательность символов. Количество символов в строке называется ее **длиной**. **Длина строки в Паскале** может лежать в диапазоне от 0 до 255. Каждый символ строковой величины занимает 1 байт памяти и имеет числовой код в соответствии с таблицей кодов ASCII.

**Строковая константа Паскаля** – последовательность символов, заключенная в апострофы.

Для описания строковых переменных в Паскале существует предопределенный тип string.

Символы в строке упорядочены, каждый из них имеет порядковый номер, начиная с первого. Имеется возможность обратиться к любому элементу строки, указав его номер, так же как это делается в одномерных массивах.

**Стандартные функции для работы со строками в Паскале**

**Copy (S, poz, n)** выделяет из строки S, начиная с позиции poz, подстроку из n символов. Здесь S – любое строковое выражение, poz, n – целочисленные выражения.

**Concat (s1, s2,...,sn)** выполняет слияние строк s1, s2,...,sn в одну строку.

**Length(S)** определяет текущую длину строкового выражения S. Результат – значение целого типа.

**Pos(subS, S)** определяет позицию первого вхождения подстроки subS в строку S. Результат – целое число, равное номеру позиции, где находится первый символ искомой подстроки. Если вхождение подстроки не обнаружено, то результат функции будет равен 0.

**Стандартные процедуры для работы со строками в Паскале**

**Delete (S, poz, n)** удаляет из строки S, начиная с позиции poz, подстроку из n символов. Здесь S – строковая переменная (в данном случае нельзя записать никакое другое строковое выражение, кроме имени строковой переменной, т.к. только с именем переменной связана область памяти, куда будет помещен результат выполнения процедуры); poz, n – любые целочисленные выражения.

**Insert(subS, S, poz)** вставляет в строку S, начиная с позиции poz, подстроку subS. Здесь subS – любое строковое выражение, S – строковая переменная (именно ей будет присвоен результат выполнения процедуры), **poz** – целочисленное выражение.

**Процедуры преобразования типов в Паскале**

**Str(x, S)** преобразует число x в строковый формат. Здесь x – любое числовое выражение, S – строковая переменная. В процедуре есть возможность задавать формат числа x. Например, str(x: 8: 3, S), где 8 – общее число знаков в числе x, а 3 – число знаков после запятой.

**Val(S, x, kod)** преобразует строку символов S в число x. Здесь S – строковое выражение, x – числовая переменная *(именно туда будет помещен результат)*, kod – целочисленная переменная *(типа integer)*, которая равна номеру позиции в строке S, начиная с которой произошла ошибка преобразования, если преобразование прошло без ошибок, то переменная kod равна 0.

При решении задач ЕГЭ работа со строками может встретиться в 27 задании. Это задание относится к повышенному уровню сложности. Недостаточно знание процедур и функций для работы со строками. Оценивается умение написания эффективной по времени и памяти программы, а значит, нельзя использовать массив, зависящий от количества входных строк.

Разберем написание эффективной программы на примере решения следующей задачи:

1. На автозаправочных станциях (АЗС) продается бензин с маркировкой 92, 95 и 98. В городе N был проведен мониторинг цены бензина на различных АЗС.

Напишите эффективную по времени работы и по используемой памяти программу (укажите используемую версию языка программирования, например, Borland Pascal 7.0), которая будет определять для каждого вида бензина, сколько АЗС продают его дешевле всего. На вход программе в первой строке подается число данных о стоимости бензина. В каждой из последующих *N* строк находится информация в следующем формате:

<Компания> <Улица> <Марка> <Цена>

где <Компания> – строка, состоящая не более, чем из 20 символов без пробелов, <Улица> – строка, состоящая не более, чем из 20 символов без пробелов, <Марка> – одно из чисел – 92, 95 или 98, <Цена> – целое число в диапазоне от 1000 до 3000, обозначающее стоимость одного литра бензина в копейках. <Компания> и <Улица>, <Улица> и <Марка>, а также <Марка> и <цена> разделены ровно одним пробелом. Пример входной строки:

Синойл Цветочная 95 2250

Программа должна выводить через пробел 3 числа – количество АЗС, продающих дешевле всего 92-й, 95-й и 98-й бензин соответственно. Если бензин какой-то марки нигде не продавался, то следует вывести 0. Пример выходных данных:

12 1 0

Информация вводится построчно. Будем читать каждую строку посимвольно, что позволит пропустить необрабатываемую информацию и считать целые числа, не используя перевод из строки в число. В данной задаче нельзя обойтись без массива, но мы будем использовать массив В, не зависящий от количества входных строк. Наш массив будет содержать элементы от 92 по 98, т.е. номер элемента массива будет являться маркой бензина, а содержать элемент массива будет число автозаправочных станций, продающий бензин по минимальной цене. Для нахождения минимальной цены возьмем ещё один массив А, тоже содержащий элементы с 92 по 98, но хранящий минимальную стоимость бензина данной марки.

*{раздел описаний}*

Var

A,b: array [92..98] of word;

C: char;

I,marka,st,n:integer;

Begin

Readln (N); *{считываем количество строк}*

For i:=92 to 98 do begin a[marka]:=3001; b[marka]:=0 end; *{заносим первоначальные значения в массив. Для массива А это 3001, т.к. будем искать минимальные значения и обнуляем массив В}*

*{Организуем цикл для считывания строк}*

For i:=1 to n do begin

 Read (c);

 While c<>’ ‘ do read (c);*{считываем и пропускаем первое слово}*

Read (c);

 While c<>’ ‘ do read (c);*{считываем и пропускаем второе слово}*

Read (marka, st)*;{считываем марку и стоимость бензина}*

If a[marka]>st then begin a[marka]:=st; b[marka]:=1 end

 else if a[marka]=st then inc (b[marka]); *{если стоимость данной марки бензина меньше минимальной, то мы её запоминаем и заносим в счетчик автостанций 1, а если стоимость равна минимальной, то увеличиваем счетчик на единицу}*

end;

writeln (b[92],’ ‘, b[95], ‘ ‘,b [98])*{выводим результат}*

end.

Аналогично решим следующие задачи:

1. После единых выпускных экзаменов по информатике в район пришла информация о том, какой ученик, какой школы сколько баллов набрал. Эта информация в том же виде была разослана в школы.

Завуч школы № 50 решила наградить двух учащихся, которые лушче всех в школе сдали информатику.

Программа должна вывести на экран имена и фамилии этих двух учеников.

Если наибольший балл набрал один человек, а следующий балл набрало несколько человек \_ нужно вывести только фамилию и имя лучшего.

Напишите эффективную программу, которая должна вывести на экран требуемую информацию. Известно, что информатику сдавало больше 5-ти учеников школы №50.

На вход программе сначала подается число учеников, сдававших экзамен. В каждой из последующих N строк находится информация об учениках в формате:

<Фамилия> <Имя> <номер школы> <баллы>, где <Фамилия> – строка, состоящая не более чем из 30 символов, <Имя> – строка, состоящая не более чем из 20 символов, где <номер школы> - целое число в диапазоне от 1 до 99, где <баллы> - целое число в диапазоне от 1 до 100. Все элемента одной строки отделены друг от друга одним пробелом. Пример входной строки:

Иванов Иван 23 57

Пример выходных данных:

Круглов Василий

Тарасова Дарья

Другой вариант выходных данных:

7

Третий вариант выходных данных:

Гусарский Илья

1. На вход программе подаются сведения о результатах олимпиады по информатике. В первой строке сообщается количество участников N, каждая из следующих N строк имеет формат: <Фамилия> <номер школы> <баллы>, где <Фамилия> – строка, состоящая не более чем из 20 символов, <номер школы> – не более чем двузначный номер, <баллы> - количество набранных баллов (целое число в диапазоне от 0 до 100). Все элемента одной строки отделены друг от друга одним пробелом. Пример входной строки:

Иванов 23 57

Требуется написать как можно более эффективную программу, которая будет выводить на экран в порядке возрастания номеров школ список, каждая строка которого содержит номер школы и максимальный набранный балл участниками олимпиады из этой школы. Номера школ, из которых не было участников, не выводятся.

*Используемые материалы:*

1. *Справочная информация по строкам (*http://www.pascal.helpov.net/index/pascal\_lines\_and\_chars\_programming*)*
2. Задачи. <http://kpolyakov.spb.ru>