

Строки

Процедуры и функции работы
со строками

Код символа и символ от кода

- **k := Ord('1'); { получили 49 }**

то же самое можно сделать с помощью преобразования типа (привести **char** к **integer**)

- **k := integer('1'); { получили 49 }**

с помощью функции **Chr** можно сделать обратный переход: получить символ по его коду, например

- **c := Chr(49); { получили символ '1' }**

то же самое можно сделать с помощью преобразования типа (привести **integer** к **char**)

- **c := char(49); { получили символ '1' }**

ФУНКЦИИ

- **n := StrToInt(s);**

преобразовать строку *s* в целое число и записать результат в переменную *n*
(*PascalABC, Delphi*)

для работы со строками в наиболее распространенных Паскаль-средах (*Turbo Pascal, Borland Pascal, PascalABC, среда АЛГО*) используют стандартные функции (здесь *s* – это переменная типа **string**, символьная строка; *n* и *r* – целые переменные)

- **n := Length(s);**

записать длину строки *s* в целую переменную *n*

- **s1 := Copy(s, 2, 5);**

записать в символьную строку *s1* подстроку строки *s*, которая начинается с символа с номером 2 и состоит из 5 символов (важно – не со 2-го по 5-ый символ!)

- **n := Pos('Вася', s);**

записать в целую переменную *n* номер символа, с которого в строке *s* начинается подстрока 'Вася' (если ее нет, в переменную *n* записывается 0); так же можно искать отдельные символы

(важно: сначала указываем, **что** ищем, а потом – **где**)

процедуры

- **Delete(s, 2, 5);**

удалить из строки *s* 5 символов, начиная со второго

- **Insert('Вася', s, 3);**

вставить в строку *s* фрагмент 'Вася', начиная с третьего символа (между 2-м и 3-м)

- **Val(s, n, r);**

преобразовать строку *s* в целое число и записать результат в переменную *n*; если при этом произошла ошибка, в переменной *r* будет номер ошибочного символа, если все нормально – ноль

Задача 1

- На вход программе подаются сведения о номерах школ учащихся, участвовавших в олимпиаде. В первой строке сообщается количество учащихся N , каждая из следующих N строк имеет формат:
- **<Фамилия> <Инициалы> <номер школы>**
- где <Фамилия> – строка, состоящая не более чем из 20 символов, <Инициалы> – строка, состоящая из 4-х символов (буква, точка, буква, точка), <номер школы> – не более чем двузначный номер. <Фамилия> и <Инициалы>, а также <Инициалы> и <номер школы> разделены одним пробелом. Пример входной строки:
- **Иванов П.С. 57**
- Требуется написать как можно более эффективную программу (укажите используемую версию языка программирования, например, Borland Pascal 7.0), которая будет выводить на экран информацию, из какой школы было меньше всего участников (таких школ может быть несколько). При этом необходимо вывести информацию только по школам, пославшим хотя бы одного участника. Следует учитывать, что $N \geq 1000$.

На что обратить внимание:

- внимательно читайте условие, убедитесь, что вы понимаете смысл каждой строчки; для каждой мелочи постарайтесь определить, зачем она добавлена в условие, что она дает для решения задачи, что ограничивает, что не разрешает делать
- определите, какая именно информация из условия нужна для решения задачи, а какая – не нужна
- определите, что именно требуется вывести на экран в результате работы программы
- начинайте составлять программу с больших блоков, записывая ее сначала на псевдокоде, а потом уточняя детали
- проверяйте «крайние» варианты (например, возможность выхода за границы массива)
- проверьте, правильно ли заданы (и заданы ли вообще) начальные значения для всех переменных
- будьте внимательны, когда в массиве есть «мертвые» элементы, которые не нужно учитывать; проверяйте, что в этом случае ваши алгоритмы (например, поиск минимального элемента) работают правильно
- проверьте, правильно ли расставлены операторные скобки **begin-end**, ограничивающие тело цикла; их обязательно нужно ставить, если в теле цикла несколько операторов
- при использовании функции **Pos** не забывайте, что первый параметр – **что** ищем (образец), а второй – **где** ищем
- чтобы эксперту было легче понять вашу программу (особенно, если она получилась «нестандартной»), пишите комментарии; объясняйте, что хранится в основных переменных
- если это возможно, желательно работать только с целыми числами; этим вы избежите проблем, связанных с округлением и неточностью хранения дробных вещественных чисел в памяти компьютера

Задача 2

- *На вход программе подаются сведения о сдаче экзаменов учениками 9-х классов некоторой средней школы. В первой строке сообщается количество учеников N , которое не меньше 10, но не превосходит 100, каждая из следующих N строк имеет следующий формат:*
- **<Фамилия> <Имя> <оценки>**,
- *где <Фамилия> – строка, состоящая не более чем из 20 символов, <Имя> – строка, состоящая не более чем из 15 символов, <оценки> – через пробел три целых числа, соответствующие оценкам по пятибалльной системе. <Фамилия> и <Имя>, а также <Имя> и <оценки> разделены одним пробелом. Пример входной строки:*
- **Иванов Петр 4 5 3**
- *Требуется написать как можно более эффективную программу (укажите используемую версию языка программирования, например, Borland Pascal 7.0), которая будет выводить на экран фамилии и имена трех худших по среднему баллу учеников. Если среди остальных есть ученики, набравшие тот же средний балл, что и один из трех худших, то следует вывести и их фамилии и имена.*

На что обратить внимание:

- в исходных данных выделите то, что не нужно для решения задачи; при чтении эти части можно просто пропускать;
- если нам не нужны фамилия и имя отдельно, можно хранить их вместе, в виде одной строки
- если нас интересует только сумма оценок, не нужно хранить их в памяти по отдельности
- если можно при решении задачи обойтись без вещественных чисел, сделав все вычисления только с целыми числами – нужно поступить именно так (иначе снимут баллы), поскольку операции с вещественными числами во многих случаях случаев выполняются неточно
- алгоритм сложности (например, сортировку) нужно использовать только тогда, когда нет алгоритма сложности ; как правило, в задачах ЕГЭ такой алгоритм всегда можно (попытаться) найти; за неэффективный алгоритм при оценке решения будут сняты баллы

За что снимают баллы:

- программа работает не для всех исходных данных, не обрабатывает некоторые частные случаи
- неверно реализован алгоритм поиска минимального элемента, сортировки и т.п.
- неэффективность алгоритма:
- используется алгоритм, имеющий сложность , когда есть алгоритм сложности
- используется несколько проходов по массиву, когда достаточно одного
- лишний расход памяти (используются дополнительные массивы или размер массива определен неверно)
- используются операции с вещественными числами, когда можно все решить в целых числах
- переменная не описана или описана неверно
- переменным не присвоены нужные начальные значения (например, не обнуляются счетчики) или присвоены неверные значения
- нет вывода результата в конце программы
- перепутаны знаки < и >, логические операции **or** и **and**
- применяется недопустимая операция, например, **div** или **mod** для вещественных чисел
- неверно расставлены операторные скобки **begin-end**
- в цикле **for** используется вещественная переменная (Паскаль)
- в цикле **while** или **repeat** не изменяется переменная цикла, из-за чего происходит закливание
- синтаксические ошибки (знаки пунктуации – запятые, точки, точки с запятой; неверное написание ключевых слов); чтобы получить 4 балла, при абсолютно верном решении нужно сделать не более одной синтаксической ошибки; на 3 балла – до трех ошибок, на 2 балла – до пяти и на 1 балл – до семи ошибок

Задача 3 (4)

- На вход программы подаются фамилии и имена учеников. Известно, что общее количество учеников не превосходит 100. В первой строке вводится количество учеников, принимавших участие в соревнованиях, N. Далее следуют N строк, имеющих следующий формат:
- **<Фамилия> <Имя>**
- Здесь *<Фамилия>* – строка, состоящая не более чем из 20 символов; *<Имя>* – строка, состоящая не более чем из 15 символов. При этом *<Фамилия>* и *<Имя>* разделены одним пробелом. Примеры входных строк:
- **Иванова Мария**
- **Петров Сергей**
- Требуется написать программу, которая формирует и печатает уникальный логин для каждого ученика по следующему правилу: если фамилия встречается первый раз, то логин – это данная фамилия, если фамилия встречается второй раз, то логин – это фамилия, в конец которой приписывается число 2 и т.д. Например, для входной последовательности
- **Иванова Мария**
- **Петров Сергей**
- **Бойцова Екатерина**
- **Петров Иван**
- **Иванова Наташа**
- будут сформированы следующие логины:
- **Иванова**
- **Петров**
- **Бойцова**
- **Петров2**
- **Иванова2**

Задача 4 (11)

- Школьная олимпиада по информатике проводилась для учеников 7-11-х классов, участвующих в общем конкурсе. Каждый участник олимпиады мог набрать от 0 до 70 баллов. Для определения призеров олимпиады сначала отбираются 25% участников, показавших лучшие результаты. Если у последнего участника, входящего в 25%, оказывается такое же количество баллов, как и у следующих за ним в итоговой таблице, все они считаются призерами только тогда, когда набранные ими баллы больше половины максимально возможных; иначе все они не считаются призерами.
- Напишите эффективную по времени работы и по используемой памяти программу, которая по результатам олимпиады будет определять минимальный балл призера олимпиады, и количество призеров было в каждой параллели (среди 7-х, 8-х, 9-х, 10-х и 11-х классов отдельно). Гарантируется, что хотя бы одного призера по указанным правилам определить можно.
- На вход программе сначала подается число участников олимпиады N . В каждой из следующих N строк находится результат одного из участников олимпиады в следующем формате:
- **<Фамилия> <Имя> <класс> <баллы>**
- где *<Фамилия>* – строка, состоящая не более, чем из 30 символов, *<Имя>* – строка, состоящая не более, чем из 15 символов, *<класс>* – число от 7 до 11, *<баллы>* – целое число от 0 до 70 набранных участником баллов. *<Фамилия>* и *<Имя>*, *<Имя>* и *<класс>*, а также *<класс>* и *<баллы>* разделены одним пробелом. Пример входной строки:
- **Семенов Сидор 11 66**
- Программа должна выводить в первой строке минимальный балл призера, а в следующей – число призеров по всем параллелям отдельно.
- Пример выходных данных:
- **63**
- **1 5 8 12 22**

Задача 5 (23)

- После единых выпускных экзаменов по информатике в район пришла информация о том, какой ученик какой школы сколько баллов набрал. По положению об экзамене каждый район сам определяет, за какой балл нужно поставить какую оценку.
- Районный методист решила, что оценку «отлично» должны получить 20% участников (целое число, с отбрасыванием дробной части). Для этого она должна определить, какой балл должен был набрать ученик, чтобы получить «отлично». Если невозможно определить такой балл, чтобы «отлично» получили ровно 20% участников, «отлично» должно получить меньше участников, чем 20%. Если таких участников не окажется (наибольший балл набрали больше 20% участников) — эти и только эти ученики должны получить «отлично».
- Напишите эффективную, в том числе и по используемой памяти, программу (укажите используемую версию языка программирования, например Borland Pascal 7.0), которая должна вывести на экран наименьший балл, который набрали участники, получившие «отлично». Известно, что информатику сдавало больше 5-ти учеников. Также известно, что есть такое количество баллов, которое не получил ни один участник.
- На вход программе сначала подаётся число учеников, сдававших экзамен. В каждой из следующих N строк находится информация об учениках в формате:
- **<Фамилия> <Имя> <Номер школы> <Количество баллов>**
- где *<Фамилия>* — строка, состоящая не более чем из 30 символов без пробелов, *<Имя>* — строка, состоящая не более, чем из 20 символов без пробелов, *<Номер школы>* — целое число в диапазоне от 1 до 99, *<Количество баллов>* — целое число в диапазоне от 1 до 100. Эти данные записаны через пробел, причём ровно один между каждой парой (то есть, всего по три пробела в каждой строке).
- Пример входной строки:
- **Иванов Иван 50 87**
- Пример выходных данных:
- **78**

Задача 6 (26)

- На вход программе подается последовательность символов, заканчивающаяся нулем. Ноль в этой последовательности единственный, среди символов обязательно есть другие десятичные цифры. Требуется написать программу, которая составляет из этих цифр число-палиндром максимальной длины (которое читается одинаково слева направо и справа налево). Если таких чисел несколько, нужно вывести минимальное из них. Нулей в числе быть не должно (ноль – признак окончания ввода). Все имеющиеся цифры использовать не обязательно, но количество цифр в ответе должно быть максимально возможным. Например, если исходная последовательность была такая:
 - **for i:=99921 downto 20**
 - то результат должен быть следующий:
 - **29192**