

*Богомолова Ольга Борисовна,
Усенков Дмитрий Юрьевич*

ЗАДАЧИ НА АНАЛИЗ И ОБРАБОТКУ ДАННЫХ: «ВЫСШИЙ ПИЛОТАЖ» ЕГЭ ПО ИНФОРМАТИКЕ

Решение задач типа С4 (в нынешних обозначениях это задачи под номером 27) требует не только умения применять типовые алгоритмы обработки данных (поиск минимума/максимума, вычисление среднего значения, определение количества элементов массива, удовлетворяющих заданному условию, сортировка и пр.), но и умений анализировать условие задачи, выделять «ключевые» данные и определять, какие данные являются излишними и не требуют запоминания, применять нестандартные приемы ввода данных, формировать индексные массивы и массивы-счетчики и т. д. Разработчики заданий ЕГЭ постоянно меняют этот тип задач, варьируя их содержание, поэтому «натаскать» детей на решение задач «С4» практически невозможно – можно только *научить* основным приемам, используемым при решении таких задач.

Описываемые далее разновидности задач «С4» использовались в ЕГЭ в течение достаточно длительного периода. И, хотя в последние два учебных года организаторы ЕГЭ «повадились» давать на экзамене задачи принципиально другого типа (которых до этого не было ни в демонстрационной версии ЕГЭ, ни в тренажах и диагностиках в течение всего предыдущего учебного года), никто не гарантирует, что прежние задачи не будут возвращены. Поэтому изучение основных приемов программирования, используемых при решении подобных задач, мы начнем именно с них.

ПРОПУСК ТЕКСТОВЫХ ДАННЫХ ИЗ ПОТОКА СИМВОЛОВ

Данные на входе программы могут представлять собой строку, содержащую несколько чисел или строковых констант, разделенных пробелом.

При чтении данных из такой входной строки оператором **read (readln)** чтение данных должно производиться слева направо с начала и до конца.

Если какие-то данные не нужны для работы программы, то для их пропуска можно использовать следующие приемы:

а) для пропуска числовых данных считать ненужные данные в отдельно объявленную «буферную» переменную соответствующего типа, которая в программе далее не используется (при чтении следующего ненужного данного используется та же переменная).

Пример:

```
...
// входная строка: <день> <месяц> <год>, где день и месяц не нужны
var Year : integer; // год
    Musor : integer; // переменная для чтения ненужных данных
...
read(Musor); // считали день
read(Musor); // считали в ту же переменную месяц
readln(Year); // считали год, переменная Year идет на обработку
```

б) для пропуска текстовых данных, поскольку их длина заранее не известна, а считывание в переменную типа **string** загружает в нее всю входную строку, нужно в цикле считывать из входной строки отдельные символы, включая пробел, завершающий очередное данное. Для пропуска еще одного такого данного повторно строится такой же цикл.

Пример:

```
...
// входная строка: <фамилия> <имя> <возраст>, где фамилия и имя не нужны
var Age : integer; // возраст
    Musor : char ; // переменная для чтения ненужного символа
...
{Пропуск фамилии}
repeat
    read(Musor)
until Musor=' ';
// считывается очередной ненужный символ, пока не окажется, что был
// считан пробел
{Пропуск имени}
repeat
    read(Musor)
until Musor=' ';
// считывается очередной ненужный символ, пока не окажется, что был
// считан пробел
{Чтение возраста}
readln(Age); // считали возраст, переменная Age идет на обработку
```

в) отдельно следует рассмотреть ситуацию, когда во входной строке нужно, например, сначала считать и обработать некоторое числовое данное, затем пропустить некоторое количество текстовых данных, а потом снова считать для обработки какое-то числовое данное. Например, если вводятся строки вида:

<номер филиала> <фамилия> <имя> <выручка> ,

где номер филиала нужно анализировать только для отслеживания нулевого номера, обозначающего конец ввода данных, а фамилия и имя не нужны.

Для чтения номера филиала можно использовать операцию **read(ID)**, где **ID** – переменная целого типа. Но тогда после чтения этого числового значения «указатель текущей позиции» в считываемой строке остановится на разделяющем пробеле между <номер филиала> и <фамилия>. Этот пробел нам нужно будет сначала отбросить, считав его как символ (**read(Musor)** вне цикла), и только после этого начинать пропуск символьных данных.

	<small><номер филиала></small>	<small><фамилия></small>	<small><имя></small>	<small><выручка></small>
Было:	1055	Иванов	Иван	10578
	^			
Стало:	1055	Иванов	Иван	10578
	^			
После				
read (Mutor) :	1055	Иванов	Иван	10578
	^			

ВАЖНО! Всегда после чтения числа (если оно во входной строке не последнее) нужно пропускать пробел после него, считывая этот пробел одиночной операцией чтения символа!

МАССИВЫ-СЧЕТЧИКИ

При определении количества элементов массива, соответствующих заданному условию (например, количества четных элементов или количества элементов, равных максимуму), мы использовали переменную-счетчик и увеличивали ее значение на 1 при обнаружении подходящего элемента.

При необходимости подсчета количеств нескольких объектов используется *массив-счетчик* размера, соответствующего количеству объектов. При обнаружении в потоке входных данных объекта, соответствующего условию, нужно увеличить на 1 значение ячейки массива-счетчика, соответствующей данному объекту.

Пример.

Для каждого из 8 филиалов фирмы (пронумерованных от 1 до 8) вразброс вводятся записи из трех чисел:

<номер филиала> <номер месяца> <прибыль/убыток>,

где прибыль выражена положительной суммой в рублях, а убыток – отрицательной. Завершение входных данных – номер филиала, равный нулю.

Требуется для каждого филиала подсчитать количество месяцев с убытком.

Решение.

1. Выделяем массив-счетчик. Количество филиалов равно 8, поэтому массив-счетчик должен состоять из 8 элементов с индексами от 1 до 8:

```
var Kol : array[1..8] of integer;
```

2. Считываем очередную строку данных (три числа в переменные **Filial**, **Mes** и **Summ**).

3. Строим цикл с предусловием:

```
while (Filial <> 0) do begin
```

4. В цикле обрабатываем вводные данные:

– если обнаружен убыток, то есть **Summ < 0**, то значение ячейки массива-счетчика, соответствующей номеру филиала, увеличиваем на 1.

```
if (Summ < 0) then Kol[Filial] := Kol[Filial] + 1;
```

5. Считываем очередную порцию данных (три числа в переменные **Filial**, **Mes** и **Summ**) – если номер филиала не равен нулю, то выполняется очередной проход цикла.

ВАЖНО! Такое построение цикла позволяет отследить ситуацию, когда входных данных нет (то есть сразу вводится завершающая строка с нулевым номером филиала).

6. После завершения цикла массив-счетчик, содержащий количество убыточных месяцев для каждого филиала, может быть дополнительно обработан (например, может быть выполнен поиск номера филиала – номера ячейки этого массива, имеющего максимальное количество убыточных месяцев).

ВАЖНО! При формировании массива-счетчика нужно не забывать, что в качестве индексов его элементов в Паскале могут использоваться не только произвольные диапазоны целых чисел, но и диапазоны символов, например букв алфавита.

ИНДЕКСНЫЕ МАССИВЫ

Возможна ситуация, когда в качестве индексов массива-счетчика невозможно (или не рационально) использовать значения, считываемые из входных данных в качестве идентификатора объектов, например:

- если вводятся словесные названия месяцев, для которых нужно выполнять подсчет (строковые данные нельзя использовать как индексы элементов массива);
- если номера объектов идут не подряд. Скажем, известно, что вводятся номера школ, для которых обрабатываются какие-то данные: 215, 386, 512, 1167, 3160, тогда при создании массива-счетчика с полным диапазоном индексов от 215 до 3160 подавляющее большинство его элементов, кроме пяти с перечисленными индексами, будут не использованы.

В этих случаях применяется *индексный массив*:

- 1) массив-счетчик объявляется по количеству обрабатываемых объектов (например, в случае с месяцами – на 12 элементов, в случае со школами – на 5 элементов);
- 2) отдельно объявляется массив того же размера, который содержит обозначения соответствующих объектов (для месяцев – их словесные названия, для школ – их номера);
- 3) при считывании входных данных соответствующее обозначение объекта ищется в индексном массиве (путем его просмотра с начала до конца); номер ячейки индексного массива, содержащей искомое обозначение объекта, запоминается;
- 4) если данный объект соответствует требуемому условию, то в массиве-счетчике значение переменной увеличивается на 1 элемент, индекс которого равен запомненному индексу элемента в индексном массиве, содержащем обозначение этого объекта.

ВАЖНО! Связь индексного массива и массива-счетчика осуществляется через общий индекс элемента. Если при выводе полученных данных требуется выводить обозначения объектов, то они берутся из индексного массива.

РАЗБОР ТИПОВЫХ ЗАДАЧ

Задача 1. Входные данные представляют собой последовательность строк – записей из базы данных студентов вуза вида:

<идентификационный номер> <фамилия> <имя> <отчество> <номер группы> <возраст>, где данные разделены одним пробелом. Завершение ввода – строка, в которой идентификационный номер равен нулю.

Требуется подсчитать количество студентов старше 25 лет. Если таких нет, то вывести сообщение: «Искомых студентов нет».

Решение

1. В данном случае в качестве счетчика может использоваться отдельная переменная:

```
var Kol : integer;
```



2. При вводе нужно вначале отдельно считывать идентификационный номер и проверять его на равенство нулю (условие завершения ввода данных). Для этого используем цикл с предусловием:

```
read(ID);
while (ID <> 0) do begin
  . . .
  read(ID);
end;
```

3. Если идентификационный номер не равен нулю, надо сначала пропустить разделяющий пробел после уже считанного идентификационного номера, а затем пропустить четыре первых данных (<фамилия> <имя> <отчество> <номер группы>). В последнем случае можно использовать как буферную переменную (для идентификационного номера и для индекса группы), так и посимвольное чтение до пробела включительно. Последний вариант является более универсальным. Для четырехкратного повторения этой операции можно использовать цикл с параметром:

```
read(c); // чтение пробела после ID
for i := 1 to 4 do begin
  repeat
    read(c)
  until c=' ';
end;
```

4. Последнее данное в каждой введенной строке считывается оператором `readln` (для перехода к следующей строке входных данных):

```
readln(Age);
```

5. Значение переменной `Age` проверяется, и если оно больше 25, то значение переменной-счетчика увеличивается на 1:

```
if Age > 25 then Kol := Kol + 1;
```

Полный текст программы:

```
Program Age_Students;
var Kol : integer;
    ID : integer;
    Age : integer;
    i : integer;
    c : char;
begin
  read(ID);
  while (ID <> 0) do begin
    read(c);
    for i := 1 to 4 do begin
      repeat
        read(c)
      until c=' ';
    end;

    readln(Age);
    if Age > 25 then Kol := Kol + 1;

    read(ID);
  end;

  if Kol = 0 then writeln('Искомых студентов нет')
  else writeln('Студентов старше 25 лет: ', Kol);
end.
```

Задача 2. Входные данные программы: подается строка английского текста. Завершение ввода: символ «точка».

Требуется подсчитать количества вхождений в нее каждой буквы от *A* до *Z* (строчные и прописные буквы не различаются) и вывести эти данные на экран (для букв, которые не встречаются в строке, не выводить ничего).



Решение

1. Требуется подсчет количества каждой из букв от *A* до *Z*. Поэтому определяем массив-счетчик с индексами элементов, обозначаемыми буквами от *A* до *Z*:

```
var Kol : array ['A'..'Z'] of integer;
```

2. Чтение входных данных – посимвольное в цикле с предусловием, пока очередной прочитанный символ не будет являться точкой:

```
read(c);
while (c <> ".") do begin
    . . .
    read(c);
end;
```

3. Обработка данных внутри цикла – проверка: если очередной символ является английской буквой, то:

- преобразовать эту букву в прописной регистр (функция **UpperCase()**);
- увеличить на 1 значение элемента массива-счетчика, индекс которого равен полученному символу.

```
if (c >= 'A') and (c <= 'Z') or (c >= 'a') and (c <= 'z') then
    Kol[UpperCase(c)] := Kol[UpperCase(c)] + 1;
```

Полный текст программы:

```
Program Alfavit;
var Kol : array ['A'..'Z'] of integer;
    c : char;
begin
    // обнуление массива-счетчика
    for c := 'A' to 'Z' do Kol[c] := 0;
    // чтение и обработка данных
    read(c);
    while (c <> '.') do begin
        if (c >= 'A') and (c <= 'Z') or (c >= 'a') and (c <= 'z') then
            Kol[UpperCase(c)] := Kol[UpperCase(c)] + 1;
        read(c);
    end;
    // вывод результата
    for c := 'A' to 'Z' do
        if Kol[c]<>0 then
            writeln('Буква ', c, ' встречается ', Kol[c], ' раз');
end.
```

Задача 3. Ученики школ города N сдавали ЕГЭ. Сведения о результатах сдачи каждого экзамена поступают в методический центр в виде наборов строк вида:

<Номер школы> <Фамилия> <Имя> <Предмет> <Оценка> ,

где оценка выражена числом баллов от 1 до 5.

Известно, что в городе 100 школ, и они имеют номера от 1100 до 1215. Всего на обработку поступило X записей, где X не превышает 1 миллион. При вводе данных вначале вводится количество записей, а затем сами записи.

Требуется подсчитать для каждой школы общее количество полученных ее учениками неудовлетворительных оценок (единиц и двоек) за все экзамены. При выводе результата – пропускать школы, ученики которых не получили ни одной неудовлетворительной оценки.



Решение

1. Требуется подсчет количеств неудовлетворительных оценок для каждой из школ с номерами от 1100 до 1215. Всего школ – 100 (то есть некоторые номера не используются). Количество неиспользуемых номеров школ равно $(1215 - 1100) - 100 = 15$. Это количество невелико, поэтому использовать индексный массив нецелесообразно. Используем массив-счетчик, индексы элементов которого меняются соответственно номерам школ от 1100 до 1215.

Каким должен быть тип этого массива? Общее число обрабатываемых записей может составлять до 1 000 000. В наихудшем случае, если все ученики получили неудовлетворительные оценки и все эти ученики относятся к одной школе, максимальное количество подсчитанных оценок (то есть максимальное значение соответствующего элемента массива) будет равно 1 000 000. Для хранения такого числа достаточно типа **integer**.

```
var Kol : array[1100..1215] of integer;
```

2. При вводе данных прежде всего считывается количество записей (переменная типа **integer**). Поэтому для чтения записей можно использовать цикл с параметром.

При чтении каждой из записей нужно:

- прочитать номер школы;
- пропустить фамилию, имя и название предмета;
- прочитать оценку.

```
read(X); // кол-во записей
for i := 1 to X do begin
  read(NoSch); // номер школы
  read(c); // считать пробел после номера школы
  for j := 1 to 3 do begin // пропуск трех данных
    repeat
      read(c)
    until c=' ';
  end;
  read(Oценка);
  . . .
end;
```

3. Проверяем: если оценка неудовлетворительная (равна 1 или 2), то увеличиваем на 1 элемент массива-счетчика, индекс которого равен номеру школы.


```
if Ocenka < 3 then
  Kol[NoSch] := Kol[NoSch] + 1;
```

ВАЖНО! Если не гарантировано, что оценка вводится правильно, то нужно проверять оценку так: `if (Ocenka = 1) or (Ocenka = 2) then ...`

Полный текст программы:

```
Program Ocenki;
var Kol : array[1100..1215] of integer;
    X, i, j, Ocenka, NoSch : integer;
    c : char;
begin
  read(X); // кол-во записей
  for i := 1 to X do begin
    read(NoSch); // номер школы
    read(c);
    for j := 1 to 3 do begin // пропуск трех данных
      repeat
        read(c)
      until c=' ';
    end;
    read(Ocenka);
    if Ocenka < 3 then
      Kol[NoSch] := Kol[NoSch] + 1;
    end;
  for i := 1100 to 1215 do // просмотр массива-счетчика
    if Kol[i]<>0 then
      writeln('Школа ', i, ' - ', Kol[i], ' учеников');
end.
```

Задача 4¹. Ученики школ города N сдавали ЕГЭ. Сведения о результатах сдачи каждого экзамена поступают в методический центр в виде наборов строк вида:

`<Номер школы> <Фамилия> <Имя> <Предмет> <Оценка>`,

где оценка выражена числом баллов от 1 до 5.

Известно, что в городе 100 школ, и они имеют номера от 1100 до 1215. Всего на обработку поступило X записей, где X не превышает 1 миллион. При вводе данных вначале вводится количество записей, а затем сами записи.

Требуется отдельно подсчитать для каждой школы количество полученных ее учениками каждой из оценок за все экзамены. При выводе результата – пропускать школы, для которых нет записей, и пропускать записи об оценках, которых нет.

Решение

1. Аналогично предыдущей задаче, нужно использовать массив-счетчик, индексы элементов которого меняются соответственно номерам школ от 1100 до 1215. Однако теперь для каждой школы надо отдельно подсчитывать количества каждой из оценок (от 1 до 5). Поэтому требуется использовать двумерный массив-счетчик, первый индекс которого меняется от 1100 до 1215, а второй индекс – от 1 до 5. Тип массива – `integer`.

```
var Kol : array[1100..1215,1..5] of integer;
```

2. При вводе данных прежде всего считывается количество записей (переменная типа `integer`). Поэтому для чтения записей используется цикл с параметром.

¹ Модификация предыдущей задачи.

- При чтении каждой из записей нужно:
- прочитать номер школы;
 - пропустить фамилию, имя и название предмета;
 - прочитать оценку.

```
read(X); // кол-во записей
for i := 1 to X do begin
  read(NoSch); // номер школы
  read(c); // считать пробел после номера школы
  for j := 1 to 3 do begin // пропуск трех данных
    repeat
      read(c)
    until c=' ';
  end;
  read(Oценка);
  . . .
end;
```

3. При обработке оценки увеличиваем на 1 элемент массива-счетчика, первый индекс которого равен номеру школы, а второй индекс равен оценке.

```
Kol[NoSch,Oценка] := Kol[NoSch,Oценка] + 1;
```

4. Вывод результатов представляет собой достаточно сложную задачу из-за требования пропуска школ, для которых нет оценок. Проще всего использовать следующий критерий:

- для очередной школы:
 - = подсчет суммы количеств каждой оценки;
 - = если эта сумма не равна нулю, то:
 - ≡ выводится номер школы;
 - ≡ в цикле по оценкам выводится информация о каждой оценке, если ее количество (элемент массива) не равно нулю;
 - = перевод строки (**writeln**);

Полный текст программы:

```
Program Ocenki_1_5;
var Kol : array[1100..1215,1..5] of integer;
    X, i, j, Oценка, NoSch, Sum : integer;
    c : char;
begin
  read(X); // кол-во записей
  for i := 1 to X do begin
    read(NoSch); // номер школы
    read(c);
    for j := 1 to 3 do begin // пропуск трех данных
      repeat
        read(c)
      until c=' ';
    end;
    read(Oценка);
    Kol[NoSch,Oценка] := Kol[NoSch,Oценка] + 1;
  end;

  for i := 1100 to 1215 do begin // просмотр массива-счетчика
    Sum := 0;
    for j := 1 to 5 do Sum := Sum + Kol[i,j];
    // если сумма оценок для данной школы равна нулю,
    // т.е. оценок для нее не задано, то для этой школы
    // ничего не выводим
```

```

if Sum<>0 then begin
  write('Школа № ', i, ':');
  for j := 1 to 5 do
    if Kol[i,j]<>0 then
      write(' ',j,' - ',Kol[i,j]);
  writeln;
end;
end;
end.

```

Задача 5. Автоматическое устройство в течение года ежедневно производит запись значения среднесуточной температуры, формируя строку вида:

<дата> <название месяца> <температура>

Всего на обработку поступило N таких строк (возможно, не все записанные устройством строки). На вход программы подается сначала число N в отдельной строке, а затем сами строки.

Требуется сформировать массив, содержащий для каждого месяца среднюю температуру за этот месяц, и вывести эти значения в формате:

<название месяца> - <средняя температура>

Если для какого-то месяца данных не было, то вывести для него строку вида:

'На <название месяца> данных нет.'

Решение

1. В данном случае требуется вычисление не просто количества, но средней температуры. При этом мы должны отдельно вычислять для каждого месяца сумму считанных значений температуры и количество таких замеров, а среднее значение можно вычислить только уже после ввода всех имеющихся записей.

Подсчет ведется для каждого из 12 месяцев, следовательно, массив-счетчик должен иметь размеры 12×2 (первый индекс – от 1 до 12 по количеству месяцев, второй индекс – от 1 до 2: первая ячейка – для подсчета суммы температур, вторая – для подсчета количества замеров). Температура может быть выражена вещественным числом, поэтому массив-счетчик должен иметь тип **real**.

Поскольку вводятся названия месяцев, то есть строки, которые не могут использоваться в качестве индексов элементов массива, требуется использовать индексный массив из 12 элементов, которые содержат соответствующие их номерам названия месяцев:

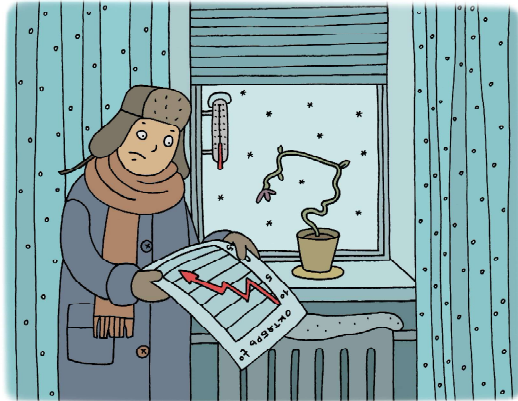
```

var IndMas : array [1..12] of string[8] := ('январь',
  'февраль', 'март', 'апрель', 'май', 'июнь', 'июль',
  'август', 'сентябрь', 'октябрь', 'ноябрь', 'декабрь');
  KolMas : array [1..12,1..2] of real;

```

2. Чтение строк производится при помощи цикла с параметром (предварительно считывается количество этих строк).

При чтении каждой строки нужно пропустить дату (число), прочитать название месяца (строка) и значение средней температуры за этот день (вещественное число). Для пропуска даты считываем ее в «буферную» переменную, позже нигде не используемую. Также не забываем считать «лишний» разделяющий пробел после только что прочитанной даты.



Особую сложность представляет чтение названия месяца. Оператор `read(Mes)`, где `Mes` – строка длиной в 8 символов (по максимальной длине названия месяца) для этого непригоден, так как считывает из входной строки строку ровно из 8 символов, тогда как названия месяцев имеют разную длину; из-за этого `Mes` после считывания будет кроме названия месяца захватывать пробел и часть цифр из значения температуры. Нам же нужно выполнить чтение в переменную `Mes` символов названия месяца до первого встреченного пробела (не включая этот пробел), поэтому используем цикл с предусловием:

```
Mes := '';  
// «обнуление» переменной (ей присваивается пустая строка)  
read(c);  
while (c<>' ') do begin  
  Mes := Mes + c;  
  read(c);  
end;
```

3. Обработка данных:

- для каждой считанной строки:
 - = просмотром в цикле от 1 до 12 индексного массива ищем индекс ячейки, значение которой совпадает с введенным названием месяца; тем самым мы определяем и запоминаем номер месяца (переменная `NoMes`);
 - = в массиве-счетчике к значению элемента `KolMas[NoMes, 1]` прибавляем считанное значение температуры;
 - = в массиве-счетчике значение элемента `KolMas[NoMes, 2]` увеличиваем на 1;
- по завершении просмотра строк для каждого значения от 1 до 12 (то есть для каждого месяца) в массиве `KolMas` выполняем вычисление средней температуры за этот месяц: если накопленное во второй ячейке количество замеров не равно нулю, то накопленную в первой ячейке сумму температур делим на это количество замеров.

```
if KolMas[i,2] <> 0 then  
  KolMas[i,1] := KolMas[i,1]/KolMas[i,2];
```

4. Вывод данных: для каждого номера месяца (от 1 до 12):

- если количество замеров за данный месяц `KolMas[NoMes, 2]` равно нулю, то выводим заданную фразу: 'На <название месяца> данных нет.';
- иначе выводим строку вида: <название месяца> - <средняя температура>.

В обоих случаях название месяца извлекается по его номеру из индексного массива: `IndMas[<номер месяца>]`.

Полный текст программы:

```
Program Mes_T;  
var IndMas : array [1..12] of string[8] := ('январь',  
  'февраль', 'март', 'апрель', 'май', 'июнь', 'июль',  
  'август', 'сентябрь', 'октябрь', 'ноябрь', 'декабрь');  
KolMas : array [1..12,1..2] of real;  
i, j, N, NoMes, Dat : integer;  
T : real;  
Mes : string[8];  
c : char;  
begin  
  read(N); // кол-во записей  
  for i := 1 to N do begin  
    read(Dat); // дата («буферная» переменная)  
    read(c); // считать разделяющий пробел после даты
```

```

Mes := '';
// «обнуление» переменной (ей присваивается пустая строка)
read(c);
while (c<>' ') do begin
    Mes := Mes + c;
    read(c);
end;
read(T);

// обработка данных
for j := 1 to 12 do
    if IndMas[j]=Mes then begin NoMes := j; break; end;
// NoMes содержит номер месяца
KolMas[NoMes,1] := KolMas[NoMes,1] + T;
KolMas[NoMes,2] := KolMas[NoMes,2] + 1;
end;

// входные данные обработаны полностью -
// вычисление средних температур за каждый месяц
for i := 1 to 12 do
    if KolMas[i,2] <> 0 then
        KolMas[i,1] := KolMas[i,1]/KolMas[i,2];

// вывод результатов
for i := 1 to 12 do
    if KolMas[i,2]=0 then
        writeln('На ',IndMas[i],' данных нет.')
    else
        writeln(IndMas[i],' - ',KolMas[i,1]);

end.

```

Задача 6. По школам района (№№ школ: 15, 181, 211, 213, 215, 367, 1115, 1125, 1560 и 1820) ежемесячно собирается информация о количестве участников и победителей олимпиад. При этом входные данные имеют вид:

<№ школы> <название месяца> <фамилия>
 <Имя> <название олимпиадного предмета>
 <статус>

где статус обозначается словом «участник» или «победитель».

Завершение ввода – строка, в которой содержится ноль.

Требуется подсчитать для каждой школы количество победителей олимпиад за весь учетный период. Для школ, в которых нет победителей, информацию не выводить.

Решение

1. Имеем 10 школ с разбросом значений их номеров от 15 до 1820. Использовать номера школ в качестве индексов массива-счетчика нерационально, поэтому необходимо использовать индексный массив, содержащий номера школ. Размер индексного массива и массива-счетчика равен 10 элементам (по количеству школ).

```

var IndMas : array [1..10] of integer := (15, 181, 211, 213, 215, 367,
1115, 1125, 1560, 1820);
    KolMas : array [1..10] of integer;

```



2. Чтение данных. Поскольку количество вводимых строк неизвестно, используется цикл с предусловием для проверки считанного значения номера школы на предмет равенства нулю (признак конца ввода).

Данные <название месяца> <Фамилия> <Имя> <название олимпиадного предмета> пропускаются.

Значение статуса ученика («участник»/«победитель») последнее по счету в строке, поэтому можно выполнить простое чтение строковой переменной.

```
read(NoSch) ;
while (NoSch <> 0) do begin
    read(c) ; // пропуск разделяющего пробела
    for i := 1 to 4 do // пропуск 4-х ненужных данных
        repeat
            read(c)
        until c=' ' ;
    read(Status) ;
    . . .
    read(NoSch) ;
end;
```

3. Обработка данных:

– просмотром в цикле от 1 до 10 индексного массива ищем индекс ячейки, значение которой совпадает с введенным номером школы; запоминаем этот индекс (переменная NoNoSch);

– проверяем считанное значение статуса: если оно равно «победитель», то в массиве-счетчике значение элемента KolMas [NoNoSch] увеличиваем на 1.

```
for i := 1 to 10 do
    if IndMas[i]=NoSch then begin NoNoSch := i; break; end;
// NoNoSch содержит номер месяца
if Status = 'победитель' then
    KolMas [NoNoSch] := KolMas [NoNoSch] + 1;
```

4. При выводе данных просматривается весь массив-счетчик, для его ненулевых ячеек выводится количество победителей, номер школы при этом берется из индексного массива.

Полный текст программы:

```
Program Olimpiads;
var IndMas : array [1..10] of integer := (15, 181, 211, 213, 215, 367,
1115, 1125, 1560, 1820);
    KolMas : array [1..10] of integer;
    NoSch, NoNoSch, i : integer;
    c : char;
    Status : string[10];
begin
    read(NoSch) ;
    while (NoSch <> 0) do begin
        read(c) ;
        for i := 1 to 4 do // пропуск 4-х ненужных данных
            repeat
                read(c)
            until c=' ' ;
        read(Status) ;
```

```

for i := 1 to 10 do
  if IndMas[i]=NoSch then begin NoNoSch := i; break; end;
  // NoNoSch содержит номер месяца
  if Status = 'победитель' then
    KolMas[NoNoSch] := KolMas[NoNoSch] + 1;
  read(NoSch);
end;

for i := 1 to 10 do
  if KolMas[i] <> 0 then
    writeln('Школа № ', IndMas[i], ' - победителей: ', KolMas[i]);
end.

```

ЗАДАЧИ ДЛЯ САМОСТОЯТЕЛЬНОГО РЕШЕНИЯ

Задача 1. Вводится строка текста, завершаемая точкой. Подсчитать в ней отдельно количества вхождений строчных латинских букв от *a* до *z*. Для букв, не входящих в заданную строку, информацию не выводить.

Задача 2. Меценаты города *N*: Петр Ведмедев, Ник Курт, Иван Корзун, Алексей Хитров, Дмитрий Митричев, Антон Хитров, Гиви Лагидзе, Джон Смирнофф, Анна Колычёва, Илья Раневский, Анатолий Петренко и Инна Румова в течение года делали пожертвования в детский фонд, в том числе неоднократно. При этом в базе данных этого фонда каждый раз делалась соответствующая запись вида:

<номер записи> <фамилия> <Имя> <месяц> <сумма взноса, руб.>

Всего таких записей накопилось *M* штук (*M* не превышает 700 000).

На вход программы подается вначале количество записей *M*, затем сами записи.

Подсчитать для каждого мецената общую сумму пожертвований и вывести эту информацию в виде:

<фамилия> <Имя> - пожертвовал <общая сумма пожертвований>

Если кто-то из меценатов пожертвовал за учетный период менее 10 000 рублей, то его фамилию и имя в список вывода не включать.

Задача 3. В городе *N* работают 15 автобаз (№№ 3, 18, 23, 25, 26, 27, 38, 52, 55, 380, 387, 500, 818, 1132 и 1508). В ГАИ города *N* при выписке штрафов в базу данных заносятся записи вида:

<фамилия> <Имя> <дата> <номер месяца> <год> <номер автобазы> <сумма штрафа, руб.>

За 5 лет таких записей накопилось *X* (где *X* не превышает 10 000).

Автотрест города *N* обрабатывает этот массив записей. При этом на вход программы вначале подается количество записей, затем сами записи.

Требуется подсчитать сумму штрафов, начисленных всем водителям каждой из автобаз в отдельности за последние 3 года (2010–2012 включительно). Если для какой-то автобазы нет начисленных ее водителям штрафов, то запись для этой автобазы не выводить.



Рекомендации по решению

Задача 1

1. Требуется подсчет количества каждой из строчных букв от *a* до *z*. Поэтому определяем массив-счетчик с индексами элементов, обозначаемыми буквами от *a* до *z*:

```
var Kol : array ['a'..'z'] of integer;
```

2. Чтение входных данных – посимвольное, в цикле с предусловием, пока очередной прочитанный символ не будет являться точкой:

```
read(c);  
while (c <> '.') do begin  
    . . .  
    read(c);  
end;
```

3. Обработка данных внутри цикла – проверка: если очередной символ является английской буквой, то увеличить на 1 значение элемента массива-счетчика, индекс которого равен полученному символу.

```
if (c >= 'a') and (c <= 'z') then    Kol[c] := Kol[c] + 1;
```

Задача 2

1. Подсчет ведется для каждого из 12 меценатов, следовательно, массив-счетчик должен иметь размер 12 элементов. Тип массива-счетчика – **integer**.

Поскольку меценаты обозначаются их фамилиями и именами, требуется использовать индексный массив из 12 пар элементов, которые содержат соответствующие каждому меценату имя и фамилию:

```
var IndMas : array [1..12,1..2] of string[10] := (('Петр',  
'Ведмедев'), ('Ник', 'Курт'), ('Иван', 'Корзун'), ('Алексей', 'Хитров'),  
('Дмитрий', 'Митричев'), ('Антон', 'Хитров'), ('Гиви', 'Лагидзе'), ('Джон',  
'Смирнофф'), ('Анна', 'Колычёва'), ('Илья', 'Раневский'), ('Анатолий',  
'Петренко'), ('Инна', 'Румова'));  
    KolMas : array [1..12] of integer;
```

2. Чтение строк производится при помощи цикла с параметром (предварительно считывается количество этих строк).

При чтении каждой строки нужно пропустить номер записи, прочитать фамилию и имя, пропустить месяц и прочитать сумму взноса.

Для пропуска даты и месяца используем посимвольное считывание до пробела включительно.

Чтение фамилии и имени выполняем путем посимвольного считывания до пробела.

3. Обработка данных считанной строки:

– поскольку для каждого из меценатов уникальным является сочетание имени и фамилии, при просмотре индексного массива нужно производить поиск по обоим этим параметрам; индекс запоминаем в переменной **NoMecen**;

– в массиве-счетчике к значению элемента **KolMas[NoMecen]** прибавляем считанное значение суммы пожертвования.

4. Вывод данных: для каждого мецената (индекс от 1 до 12):

– если сумма пожертвований превышает 10000, то выводим строку вида:

```
<фамилия> <Имя> - пожертвовал <общая сумма пожертвований>
```

При этом фамилия и имя извлекаются из индексного массива.

Задача 3

1. Имеем 15 автобаз с большим разбросом значений их номеров, поэтому необходимо использовать индексный массив, содержащий номера автобаз. Размер индексного массива и массива-счетчика равен 15 элементам (по количеству автобаз).

```
var IndMas : array [1..15] of integer := (3, 18, 23, 25, 26, 27, 38, 52,
55, 380, 387, 500, 818, 1132, 1508);
    KolMas : array [1..15] of integer;
```

2. Чтение строк производится при помощи цикла с параметром (предварительно считывается количество этих строк).

При чтении каждой строки нужно пропустить данные: <Фамилия> <Имя> <дата> <номер месяца>.

Прочитываем год, номер автобазы и сумму штрафа.

3. Обработка данных:

– просмотром в цикле от 1 до 15 индексного массива ищем индекс ячейки, значение которой совпадает с введенным номером автобазы; запоминаем этот индекс (переменная **NoNoAvt**);

– проверяем считанное значение года: если он равен 2010, 2011 или 2012, то в массиве-счетчике к значению элемента **KolMas [NoNoAvt]** прибавляем значение **Sum**.

4. При выводе данных просматривается весь массив-счетчик, для его ненулевых ячеек выводится накопленная сумма штрафов, номер автобазы при этом берется из индексного массива.

(Продолжение следует.)

Богомолова Ольга Борисовна,
доктор педагогических наук,
почетный работник сферы
образования Российской Федерации,
Заслуженный учитель города
Москвы, учитель информатики
и математики ГБОУ СОШ № 1360,
г. Москва,

Усенков Дмитрий Юрьевич,
ГБОУ СОШ № 1360, г. Москва.

