

# Песни о Паскале

## **Задачи секции «А слабо?»**

редакция 12.5

от 2013-04-01

## Аннотация

В этом файле представлены задачи из секции «А слабо» книги «Песни о Паскале». Многие из этих задач годятся при обучении и другим языкам программирования.

# Оглавление

|               |    |
|---------------|----|
| Глава 5.....  | 5  |
| Глава 7.....  | 6  |
| Глава 8.....  | 7  |
| Глава 9.....  | 8  |
| Глава 10..... | 9  |
| Глава 11..... | 10 |
| Глава 12..... | 11 |
| Глава 13..... | 12 |
| Глава 14..... | 13 |
| Глава 15..... | 14 |
| Глава 16..... | 15 |
| Глава 17..... | 16 |
| Глава 18..... | 18 |
| Глава 19..... | 19 |
| Глава 20..... | 20 |
| Глава 23..... | 21 |
| Глава 24..... | 22 |
| Глава 25..... | 23 |
| Глава 26..... | 24 |
| Глава 27..... | 25 |
| Глава 29..... | 26 |
| Глава 30..... | 27 |
| Глава 31..... | 28 |
| Глава 32..... | 29 |
| Глава 33..... | 30 |
| Глава 34..... | 32 |
| Глава 35..... | 33 |
| Глава 36..... | 34 |
| Глава 37..... | 35 |
| Глава 38..... | 36 |
| Глава 39..... | 38 |
| Глава 40..... | 39 |
| Глава 41..... | 40 |
| Глава 42..... | 41 |
| Глава 43..... | 42 |
| Глава 44..... | 43 |
| Глава 45..... | 45 |
| Глава 46..... | 46 |
| Глава 47..... | 47 |
| Глава 48..... | 49 |
| Глава 49..... | 50 |
| Глава 50..... | 51 |
| Глава 51..... | 52 |
| Глава 52..... | 53 |
| Глава 53..... | 55 |
| Глава 54..... | 56 |
| Глава 55..... | 57 |
| Глава 56..... | 58 |
| Глава 57..... | 59 |
| Глава 58..... | 60 |
| Глава 59..... | 62 |

|               |    |
|---------------|----|
| Глава 60..... | 63 |
| Глава 61..... | 64 |

## Глава 5

**А)** Найдите ошибки в следующей программе.

```
begin  
  
Writeln(ПрЫветик!)  
  
end
```

Сначала сделайте это в уме, а затем на компьютере. Объясните, почему компилятор не нашел ошибки в слове «ПрЫветик». Или слабо?

**Б)** Будет ли работать следующая программа?

```
begin Writeln('Begin End.') end.
```

**В)** Попробуйте написать программу, выводящую на экран не одну, а две строки, например.

```
Без труда  
  
Не выловишь калошу из пруда
```

Здесь нужны две процедуры печати, следующие друг за другом. Между процедурами должен стоять специальный разделитель – точка с запятой (;).

## Глава 7

**А)** В нашей последней программе остался маленький изъян. Со временем вы можете забыть о том, что для завершения программы надо нажать клавишу «Enter». Пусть программа сама напомнит об этом, печатая после приветствия напоминание:

|  |
|--|
| Для завершения программы нажмите Enter |
|--|

Внесите это изменение в программу. Или слабо?

**Б)** Измените программу так, чтобы в каждой строке разместилось по два оператора. Откомпилируйте и проверьте программу в действии. Изменилось ли что-то в её поведении?

**В)** Нарисуйте блок-схемы, поясняющие, как вы обычно проводите свой будний и выходной день.

## Глава 8

**А)** Что напечатает следующая программа, если ваша любимая команда – «Спартак»?

```
const Champ = ' - чемпион!';  
var Team : string;  
begin  
  Writeln('Ваша любимая команда?');  
  Readln(Team);  
  Writeln(Team, Champ);  
  Readln  
end.
```

**Б)** Найдите (и исправьте, если можно) ошибки в следующих программах.

```
begin  
const Pele = 'Эдсон Арантес ду Насименту';  
Writeln('Лучший футболист мира - ', Pele);  
Readln  
end.
```

```
begin  
Writeln('Как тебя зовут?');  
var Name : string;  
Readln(Name);  
Writeln('Здравствуй, ', Name);  
Writeln('Нажми Enter'); Readln;  
end.
```

```
const Pele = 'Эдсон Арантес ду Насименту';  
begin  
Writeln('Лучший футболист мира');  
Readln(Pele);  
Writeln(Pele);  
Readln  
end.
```

## Глава 9

**А)** Что напечатает следующая программа?

```
const Pele = 'Эдсон Арантес ду Насименту';  
begin  
Writeln('Pele = ', Pele);  
end.
```

**Б)** А эта программа что напечатает?

```
var A, B : string;  
begin  
A:='12';    B:='34';  
Writeln('A+B= ', A+B);  
end.
```

**В)** Является ли следующий оператор оператором присваивания?

```
const Pele = 'Эдсон Арантес ду Насименту';
```

**Г)** Пусть ваша программа запросит у пользователя его адрес, а именно: город, улицу, номер дома и номер квартиры. А затем напечатает адрес одной строкой в таком виде:

```
Город: ГГГ  Улица: УУУ  Дом: ДДД  Квартира: ККК
```

Сделайте два варианта программы: один – с печатью нескольких параметров оператором **Writeln**, другой – с объединением строк.

**Д)** Какие из следующих операторов «забракует» компилятор?

```
const Pele = 'Эдсон Арантес ду Насименту';  
      ABBA : string = 'Музыкальный шедевр из Швеции';  
var    Moscow : string;  
begin  
      Pele  := 'Лучший футболист мира';  
      ABBA  := 'Распевают частушки';  
      Moscow:= 'Столица олимпиады';  
end.
```



## Глава 10

**А)** В программах для «часового» укажите начало и конец условного оператора (то есть, первый и последний его символ, включая вложенные операторы).

**Б)** Напишите программу, которая спрашивает, идет ли дождь, и на ответ «да» выводит сообщение «А зонта то у тебя нет!».

## Глава 11

**А)** Сколько операторов можно поместить в операторном блоке?

**Б)** Найдите ошибку в этом кусочке программы, проверьте свое решение на компьютере.

```
Writeln('Что дождь? Идет?'); Readln(S);  
if S = 'ara' then  
    begin  
        Writeln('А зонтик ты так и не купил!');  
        Writeln('Сколько раз напоминать?');  
    end;  
else begin  
        Writeln('На этот раз тебе повезло!');  
    end;
```

## Глава 12

- А)** Сколько операторов можно вставить между **REPEAT** и **UNTIL**?
- Б)** Будет ли проверяться условие в **UNTIL** при досрочном выходе из цикла?
- В)** Возьмите за основу программу «P\_11\_1» и сделайте из нее циклический вариант. Или слабо?
- Г)** Напишите программу для угадывания слова. Она должна запрашивать от пользователя строки, пока тот не введет слово, предусмотренное в программе.

## Глава 13

**А)** Что будет напечатано в результате выполнения следующего фрагмента? Проверьте себя на компьютере.

```
S:='123';  
Writeln (S='123');
```

**Б)** Переведите на русский язык следующее выражение на Паскале.

```
if (S='') and (A or B) then ...
```

**В)** Напишите программу к бортовому компьютеру для плана проезда, показанного на рис. 1. Или слабо?

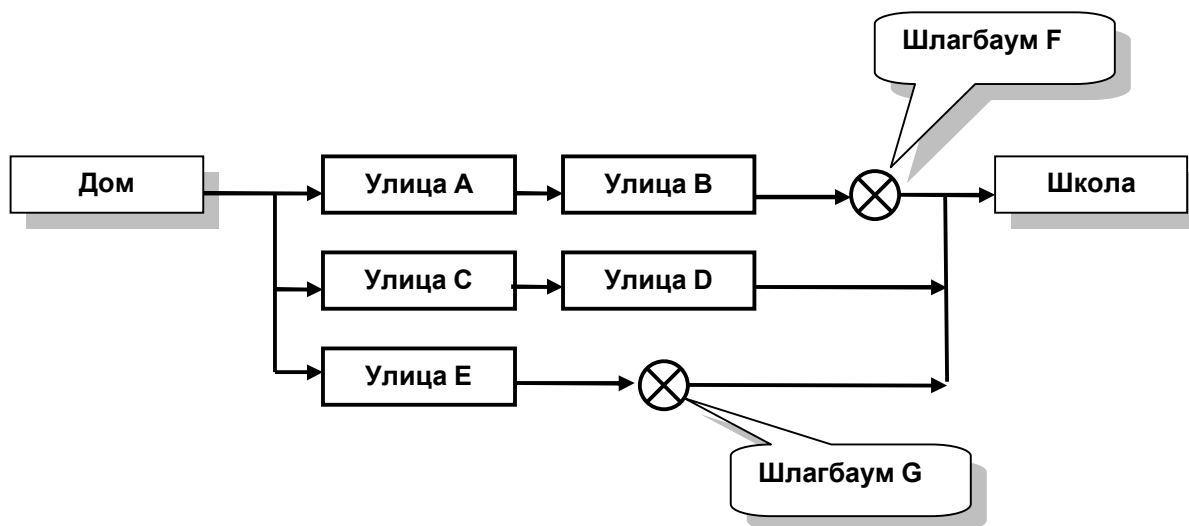


Рис. 1 – Схема проезда к задаче «В»

**Г)** В переменные **m1**, **m2** и **m3** вводится итог подбрасывания трех монет так, чтобы **TRUE** соответствовал «орел», а **FALSE** – «решка». Составить пять выражений таких, чтобы они выдавали **TRUE** для следующих случаев:

- у всех монет выпал «орел»;
- у всех монет выпала «решка»;
- все монеты упали одинаково;
- у первой – «решка», у прочих – «орел»;
- у первой – «орел», а две остальные упали одинаково.

Подсказка: логические данные можно сравнивать; сравнение обладает самым низким приоритетом, и потому внутри выражений заключается в скобки, например: **m1 and (m2=m3)**.

## Глава 14

**А)** Найдите ошибки в следующих операторах и объясните их.

```
var   N, M : integer;  
      S : string;  
begin  
      N:= '10';  
      S:= N + 5;  
      M:= S - 1;  
      if S=N then;  
end.
```

Проверьте свои догадки, призвав на помощь компилятор.

**Б)** Перепишите программу «P\_14\_1», не прибегая к процедуре **Break**. В чем, по-вашему, слабость этого второго варианта? Можно ли обойтись в программе «P\_14\_1» без булевой переменной **R** и строковой **S**? Напишите такой вариант программы. Или слабо?

**В)** Пусть ваша программа запросит три числа: **A**, **B** и **C**, а затем напечатает большее из трех чисел.

Подсказка: примените булевы выражения вкупе с операциями сравнения, которые в булевых выражениях надо заключать в скобки, например:

```
if (A>=B) and (A>=C) then . . .
```

Скобки ставятся по той причине, что булевы операции приоритетней операций сравнения, а нам нужно выполнять сравнения раньше. Подробнее о булевых операциях с числами сказано в главе 48.

**Г)** В стене прорублено прямоугольное отверстие со сторонами **A** и **B**. Пусть ваша программа определит, пройдет ли в него кирпич с ребрами **X**, **Y**, **Z**.

**Д)** Площадь земельного участка вычисляется как произведение его сторон: ширины **A** и длины **B**. Введите в программу ширину и длину двух участков (**A1**, **B1** и **A2**, **B2**), и пусть программа напечатает ширину и длину большего из них.

## Глава 15

**А)** В каких пределах будут генерироваться числа следующими выражениями:

`10+Random(10) ;`

`Random(20) ;`

`Random(10) + Random(10) ;`

Проверьте себя на компьютере!

**Б)** Сколько чисел будет напечатано следующей программой? Испытайте на практике.

```
var x : integer;
begin
    repeat
        x := Random(20) ;
        Writeln(x) ;
    until x=1;
end.
```

**В)** А если в начале предыдущей программы вставить **Randomize**? Можно ли предсказать результат? Или слабо?

**Г)** Найдите способ сформировать ряд случайных булевых значений (**False**, **True**), напечатайте 20 из них. Подсказка: булевы значения можно получить, сравнивая два случайных целых числа.

**Д)** Сгенерируйте два случайных числа (в диапазоне от 1 до 10) так, чтобы они не совпадали. Сделайте то же самое для трех чисел.

## Глава 16

**А)** Какой ответ будет выпадать чаще других, если условием в операторе **CASE** нашей программы поставить выражение **Random(100)**?

**Б)** Напишите программу, которая бы запрашивала номер дня недели, и в ответ печатала бы название этого дня («понедельник», «вторник» и так далее).

**В)** Пусть пользователь введет число – свой возраст в годах. Ваша программа должна напечатать фразу: «Вам столько-то лет» с правильным окончанием, например: «Вам 20 лет», или «Вам 34 года», или «Вам 41 год». Подсказка: надо определить последнюю цифру года операцией **MOD 10**. Некоторые числа выпадают из общего правила, их надо проверить особо (например, 11, 12, 13, 14).

**Г)** Пользователь вводит число – номер месяца от 1 до 12, а программа должна сообщить соответствующее ему время года: зима, весна, лето, осень. Подсказка: в одной ветви можно применить несколько меток, например.

```
case N of
  1, 2, 12 : Writeln('Зима');
```

**Д)** Танк в компьютерной игрушке может двигаться в одном из четырех направлений, обозначим их числами: 1 – север, 2 – восток, 3 – юг, 4 – запад. Направление движения изменяется тремя командами: 1 – направо, 2 – налево, 3 – кругом. Пользователь вводит начальное направление движения, а затем ряд команд. Программа должна определять и печатать всякий раз новое направление. Выход из цикла – команда 0.

**Е)** Исходное состояние шахматных фигур известно всякому (если вы исключение из правила, ознакомьтесь с основами шахмат). Пользователь в цикле вводит число, по которому программа печатает название фигуры, стоящей на соответствующей вертикали шахматной доски (от 1 до 8). Ноль служит для выхода из цикла, а на все прочие числа программа сообщает об ошибке.

**Ж)** Программа запрашивает в цикле два числа: вертикаль и горизонталь шахматной доски (числа от 1 до 8), а затем печатает цвет клетки на их пересечении. Если хотя бы одно из чисел равно нулю, цикл завершается. Если числа выходят за указанные пределы, сообщает об ошибке и повторяет запрос чисел.

**Примечание:** на пересечении 1-й строки и 1-го столбца находится чёрная клетка.

## Глава 17

**А)** Дайте ученику возможность отказаться от сдачи экзамена. Признаком отказа будет ввод числа ноль в качестве ответа. В этом случае надо досрочно выйти из цикла и обойти выставяющий оценку оператор.

**Б)** Напишите программу, которая по введенному числу дает заключение о том, какому дню недели оно соответствует – рабочему (1-5) или выходному (6,7), например:

```
День = 2  
Рабочий  
День = 7  
Выходной  
День = 20  
Ошибка!
```

Здесь подчеркнутые числа напечатаны пользователем.

**В)** Напишите программу, которая, запросив число **N**, печатала бы числа от 1 до **N** в обратном порядке, например:

```
N = 3  
3  
2  
1
```

**Г)** Существует вариант оператора **FOR...**, где счетчик цикла не наращивается, а уменьшается, этот оператор записывается так:

**FOR** N:= начальное\_значение **DOWNTO** конечное\_значение **DO** оператор

Ключевое слово **DOWNTO** задает счет в обратном порядке (**DOWN** – «вниз»); при этом предполагается, что начальное значение счетчика больше или равно конечному, иначе цикл не выполнится ни разу. Воспользуйтесь этим оператором для решения предыдущей задачи (**В**).

**Д)** Пусть программа запросит два числа **N** и **M**, а затем вычислит их произведение без использования операции умножения (\*). Подсказка: организуйте цикл суммирования **N** раз числа **M**.

**Е)** Напишите программу, вычисляющую сумму чисел от 1 до **N**, где **N** – число, вводимое пользователем.

**Ж)** Напишите программу, вычисляющую сумму только тех чисел от 1 до **N**, которые делятся либо на три, либо на пять.



**И)** Платный участок трассы протянулся с **P1** до **P2** километра (**P1 < P2**). А пост ГАИ размещен на километре **M**. Расположен ли этот пост на платном участке трассы? Пусть ваша программа разберется с этим.

**К)** Дорожная служба запланировала ремонт трассы на участке с **R1** по **R2** (**R1 < R2**). В сочетании с условием предыдущей задачи ваша программа должна определить:

- Будут ли ремонтировать весь платный участок **P1-P2** ?
- Будут ли ремонтировать хотя бы часть платного участка **P1-P2** ? Если да, то определить длину ремонтируемой платной части.
- Будут ли ремонтировать хотя бы часть бесплатного участка? Если да, то определить длину ремонтируемой бесплатной части.

## Глава 18

**А)** Напишите программу для подсчета букв «А» во введенной пользователем строке. Или слабо?

**Б)** Напишите программу, меняющую символы «А» строки на символы «Б». Подсказка: присвоение значения некоторому символу строки делается оператором вида **S[i] := ...**

**В)** Что делают со строкой **S** следующие операторы?

```
for i:=1 to Length(S) do S:= S + S[i];  
for i:=Length(S) downto 1 do S:= S + S[i];
```

Проверьте свои предположения на практике.

**Г)** Записи телефонных номеров обычно содержат дополнительные символы (скобки, черточки, пробелы), - чтобы удобней запоминались. Например: 8 (123) 45-67-89. Предположим, что пользователь их так и вводит. Пусть ваша программа преобразит строку с номером, удалив из нее все символы, кроме цифр. Например, после ввода указанного выше номера она должна напечатать: 8123456789.

**Д)** Пусть ваша программа напечатает введенную пользователем строку вразрядку, добавляя пробел после каждого символа, например: **'Pascal' → 'P a s c a l'**.

## Глава 19

**А)** Напишите еще одну версию процедуры **Pause**, выводящую сообщение либо на русском, либо на английском языке. Параметр этой процедуры должен быть булевым и работать она должна так.

```
Pause(true);      { печатается «Нажмите Enter...» }  
Pause(false);     { печатается «Press Enter...» }
```

**Б)** Напишите и испытайте процедуру (назовем её **Line** – «линия»), печатающую строку заданной длины, составленную из звездочек, например.

```
Line(3);          { печатает «***» }  
Line(7);          { печатает «*****» }
```

Подсказка: внутри процедуры надо организовать цикл.

**В)** Напишите процедуру для очистки экрана, она может пригодиться вам в будущем. Подсказка: можно напечатать несколько десятков пустых строк (не менее 25, - зависит от настройки консольного окна).

**Г)** Напишите процедуру, принимающую два числа (два параметра), и печатающую два числа: их сумму, и их разность.

### Задачи на темы предыдущих глав

**Д)** Пользователь вводит строку с телефонным номером (только цифры), количество цифр заранее неизвестно. Ваша программа должна дополнить номер дефисами, разбивающими его на триады, т.е. по три цифры двумя способами:

- начиная с первых цифр, например 112-345-1;
- начиная с последних цифр, например 1-123-451.

**Е)** Почтальон разносит газеты по улице, состоящей из **N** домов. Четные и нечетные номера расположены по разные стороны улицы. В здравом уме почтальон не рискует лишний раз переходить её. Ваша программа должна напечатать последовательность номеров, по которым будут разнесена почта, когда почтальон начинает работу:

- с первого дома;
- со второго дома;
- с N-го (последнего) дома.

## Глава 20

**А)** В 17-й главе нами создан экзаменатор, проверяющий знания таблицы умножения. Переработайте программу `P_17_1` так, чтобы операторы, выставляющие оценку, были выделены в процедуру, принимающую один параметр – количество допущенных ошибок.

**Б)** Создайте процедуру, печатающую все числа, кроме единицы, на которые без остатка делится число **N**, где **N** – параметр процедуры. Напишите программу для проверки этой процедуры.

**В)** Два сотрудника подали своему начальнику заявления на отпуск. Первый попросил отпустить его с **A1** по **B1** день (дни отсчитываются с начала года), второй – с **A2** по **B2** день. Предполагается, что **A1 < B1** и **A2 < B2**. Однако дело требует, чтобы кто-то из сотрудников находился на рабочем месте. Мало того, при смене отдыхающих необходимо не менее 3-х дней их совместной работы – для передачи дел. Напишите программу с процедурой, принимающей четыре указанных выше параметра, и печатающую заключение о том, удовлетворяют ли заявления работников требованиям начальника.

**Г)** Подойдя к перекрестку, пешеход думает о том, переходить ли ему улицу, или остановиться. На решение влияет характер пешехода и еще два фактора: сигнал светофора и близость опасно движущегося транспорта. Напишите программу с процедурой, которая принимает и печатает решение в зависимости от переданных в неё трех параметров, а именно.

- Параметр **A = true**, если горит зеленый;
- Параметр **B = true**, если поблизости опасно движется транспорт;
- Параметр **C** – это число, определяющее характер пешехода так:
  - 1 - послушный и осторожный – учитывает светофор и опасность;
  - 2 - послушный, но беспечный – смотрит только на светофор;
  - 3 - хитрый вольнодумец – идет только на красный, если это ничем не грозит;
  - 4 - непримиримый вольнодумец – идет только на красный, невзирая ни на что;
  - 5 - «экстремал» – идет только на красный, но так, чтобы грозила опасность;
  - 6 - «безбашенный» – идет, несмотря ни на что;
  - 7 - запуганный – никогда не идет через дорогу, а ищет подземный переход.

## Глава 23

**А)** Напишите функцию для поиска буквы в заданной строке. Она должна возвращать **TRUE**, если в строке есть хоть одна эта буква, и **FALSE** в противном случае. Напишите программу для проверки функции. Или слабо?

**Б)** Напишите функцию для определения позиции буквы в заданной строке. Функция должна вернуть позицию первой такой буквы или ноль, если буквы в строке нет. Напишите программу для проверки функции.

**В)** Напишите функцию и программу для её проверки, принимающую число и возвращающую строку: слово «четное» или «нечетное» в зависимости от четности или нечетности параметра. Подсказка: для проверки четности числа **N** надо проверить остаток от его деления на два: **if (N mod 2) = 0 then ...**

## Глава 24

**А)** Переработайте программу шифрования так, чтобы ключ задавался пользователем с клавиатуры (в начале работы программы). Затем этот ключ должен передаваться в процедуры и функции через параметр. Заголовки процедур и функций должны быть такими.

```
function CryptChar(arg: char; key: integer): char;  
procedure CryptString(var arg: string; key: integer);
```

**Б)** Предположим, вы зашифровали строку пять раз подряд. Можно ли расшифровать её? И как это сделать?

**В)** Для введенной пользователем строки напечатать позиции всех встречающихся в ней символов, кроме пробелов, в алфавитном порядке. Для символов, которые встречаются несколько раз, напечатать позиции в одной строке. Например, для слова «PASCAL»:

```
A - 2 5  
C - 4  
L - 6  
P - 1  
S - 3
```

**Г)** Для введенной пользователем строки напечатать позиции всех встречающихся в ней символов, кроме пробелов, в порядке их следования в строке. Например, для слова «PASCAL»:

```
P - 1  
A - 2 5  
S - 3  
C - 4  
L - 6
```

**Д)** Строки текстовых файлов порой содержат управляющие символы, например символ горизонтальной табуляции (код 9). Шифрование этих символов нарушит структуру файла. Исправьте функции **Crypt** и **DeCrypt** так, чтобы они не изменяли символы, коды которых меньше 32.

## Глава 25

**А)** Можно ли связать текстовую переменную **F** с файлом оператором присваивания?

|                                       |
|---------------------------------------|
| <code>F := 'c:\autoexec.bat' ;</code> |
|---------------------------------------|

**Б)** Напишите программу для вывода на экран файла, имя которого задается с клавиатуры.

**В)** Напишите три функции для подсчета:

- количества строк в файле;
- количества видимых символов в файле;
- количества всех символов файла (объем файла).

Функции принимают один параметр – ссылку на файловую переменную. Напишите программу, подсчитывающую упомянутые выше характеристики файла.

**Г)** Объявите две файловые переменные, свяжите их с одним и тем же файлом, а затем откройте через обе переменные. Вызовет ли это ошибку? Объясните результат, исходя из здравого смысла.

**Д)** Усовершенствуйте программу «вопрос-ответ» (глава 16) с тем, чтобы ответы хранились не в программе, а в отдельном текстовом файле. Тогда пользователи программы сами смогут сочинять ответы.

**Е)** Напишите процедуру для вывода на экран **n**-й строки файла, где **n** – параметр процедуры. Воспользовавшись этой процедурой, напишите программу для распечатки строк файла в обратном порядке. Подсказка: предварительно посчитайте количество строк в файле.

## Глава 26

**А)** Напишите программу, которая создает файл, печатает в него несколько строк с числами, а затем выводит этот файл на экран. Воспользуйтесь одной файловой переменной.

**Б)** Напишите программу для нумерации строк файла. Строки исходного файла должны копироваться в конечный файл с добавлением перед каждой строкой её номера, например.

Исходный файл:

```
В лесу родилась елочка,  
В лесу она росла.  
Зимой и летом стройная,  
Зеленая была.
```

Конечный файл:

```
1  
В лесу родилась елочка,  
2  
В лесу она росла.  
3  
Зимой и летом стройная,  
4  
Зеленая была.
```

**В)** Скопировать один файл в другой:

- с перестановкой местами четных и нечетных строк;
- с перестановкой строк в обратном порядке (см. условие задачи «Е» к 25-й главе).

**Г)** Для передачи по интернету секретного текстового файла создать из него два других: в первый записать нечетные строки исходного файла, а во второй – четные. Напишите программу для этого, или слабо?

**Д)** Создать программу для объединения двух файлов: из первого составляются нечетные строки конечного файла, а из второго – четные (см. условие предыдущей задачи).



## Глава 27

**А)** Выясните, какой код ошибки возвращает функция **IOResult** при попытке открыть для чтения несуществующий файл. Напишите для этого небольшую программу.

**Б)** Сделайте то же самое для случая, когда программа пытается открыть для записи файл, уже открытый вами в редакторе текста.

**В)** Некоторые строки исходного файла содержат круглые скобки (это может быть программа или математические выкладки – неважно). Ваша программа должна распечатать строки, где скобки расставлены неверно, вот примеры.

|           |               |
|-----------|---------------|
| 2+3       | – правильно ; |
| (2+3      | – ошибка ;    |
| () 2+3 () | – правильно ; |
| )) 2+3 (( | – ошибка .    |

Рекомендация: для исследования строки напишите булеву функцию **Check**, возвращающую **TRUE**, если скобки расставлены без ошибок.

**Г)** Дребезг контактов – с этим явлением борются специалисты по электронике. Дребезг возникает при замыкании-размыкании кнопок, тумблеров, реле и других подобных устройств. Сигнал от контактов поступает в микропроцессор с некоторой периодичностью, скажем, 100 раз в секунду. Если контакт разомкнут, микропроцессор принимает «0», а если замкнут – «1». В ходе замыкания-размыкания контакт неустойчив, и процессор получает несколько чередующихся нулей и единиц, – его программа должна отфильтровать эти ложные срабатывания.

Ваша программа будет моделировать поведение микропроцессора. Входной файл содержит последовательность нулей и единиц (по одному символу в строке). Из первой строки берется исходное значение сигнала, а дальше сигнал на выходе программы должен формироваться так: если три подряд идущие значения совпадают, то берется это новое значение, а иначе сохраняется текущее, например:

| На входе | На выходе |
|----------|-----------|
| 0        | 0         |
| 1        | 0         |
| 0        | 0         |
| 1        | 0         |
| 1        | 0         |
| 1        | 1         |
| 0        | 1         |

Выходной файл должен содержать две колонки: входной и выходной сигналы.

## Глава 29

**А)** Напишите программу для преобразования второго варианта базы данных «Police.txt» (с несколькими числами в строке) в первый вариант (по одному числу в строке). Или слабо?

**Б)** Можно ли при решении предыдущей задачи назначить одно и то же имя как входному, так и выходному файлам? Испытайте свое решение.

## Глава 30

**А)** Функция **Trunc** выделяет целую часть вещественного числа, например.

```
Writeln (Trunc( 12.345 ));    { 12 }
```

Исследуйте её и придумайте способ выделения дробной части вещественного числа. Напишите подходящую функцию и программу для её проверки.

**Б)** Объясните и проверьте, что напечатает следующая программа.

```
var N: integer;  
begin  
    for N:=1 to 20 do Writeln ( ' ':N, N );  
end.
```

**В)** Сформируйте файл «Numbers.txt», поместив в него 100 случайных чисел в диапазоне от 0 до 999 (некоторые числа могут повторяться). Затем найдите в этом файле: 1) максимальное и минимальное число; 2) сумму всех чисел; 3) среднее арифметическое – напечатайте его с точностью 2 знака после точки.

**Г)** Сканирование марсианской поверхности дало файл, содержащий высоту отдельных его точек вдоль одного из направлений, – пусть это будет файл «Numbers.txt» из предыдущей задачи. Найдите точки, где вероятней всего обнаружить марсианскую воду, – на следующем ниже рисунке они обозначены буквами **W**. Программа должна напечатать две колонки: порядковый номер точки относительно начала файла (счет от нуля) и высоту точки (такие точки математики называют локальными минимумами).

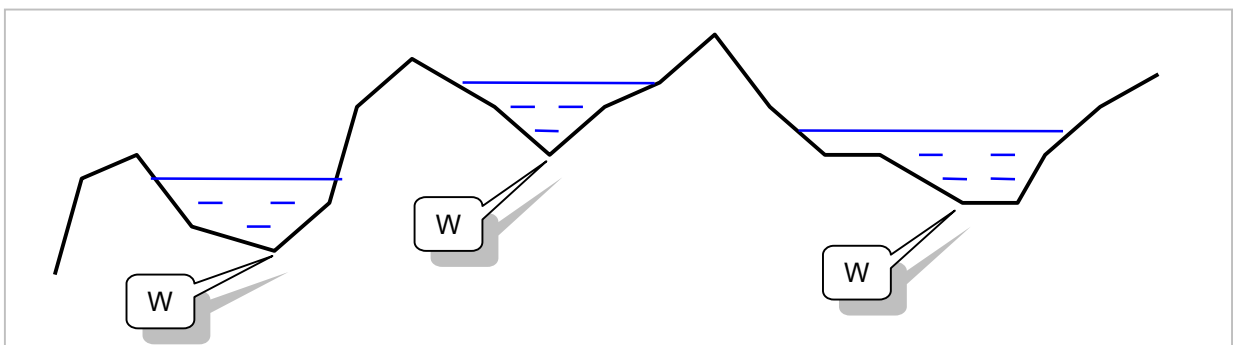


Рис. 2 - Рельеф марсианской поверхности

## Глава 31

**А)** Напишите программу для преобразования первого варианта базы данных «Police.txt» (которая содержит по одному числу в строке) во второй вариант (будет содержать по три числа в строке).

**Б)** Файл с физическими данными старшеклассников содержит три колонки: фамилия, рост и вес ученика. Создайте программы для решения следующих задач:

- отбор кандидатов для занятий баскетболом, – рост кандидата должен составлять не менее 175 см;
- поиск учеников с избыточным весом, для которых разница между ростом ученика (см) и его весом (кг) составляет менее 100.

Ваши программы должны сформировать соответствующие файлы с фамилиями и данными учеников.

## Глава 32

**А)** Напомню, что функция **SizeOf** возвращает объем памяти, который занимает переменная или тип данных, например.

```
Writeln( SizeOf( LongInt ) ); { 4 }  
Writeln( SizeOf( M1 ) );      { 1 }
```

Воспользуйтесь ею для распечатки размеров всех известных вам порядковых типов данных.

**Б)** Перечислимые типы и диапазоны строятся на базе других типов данных (**Byte**, **ShortInt** и так далее). Какие типы данных, по вашему мнению, будут положены в основу следующих диапазонов:

```
var  N : -10..10;  
     M : -200..200;  
     R : 0..40000;  
     L : 0..400000;  
     S : '0'..'9';
```

**В)** Процедура печати **Writeln** не способна распечатать название месяца, представленного в перечислении. Придумайте, как это сделать, и напишите процедуру для этого (воспользуйтесь оператором **CASE**).

**Г)** «Не думай о секундах свысока...». Штирлицу подарили секундомер, который показывал секунды, прошедшие с начала суток. Пусть ваша программа переведет это число в привычные часы, минуты и секунды. Вот две подсказки: во-первых, воспользуйтесь операциями **DIV** и **MOD**. Во-вторых, объявляя переменную для секунд, примените тип **LONGINT** (а не **INTEGER**), поскольку количество секунд в сутках (86400) не поместится в типе **INTEGER**.

## Глава 33

**А)** Напишите две функции, округляющие вещественное число:

- до большего значения (например:  $3.1 \rightarrow 4$ ;  $3.9 \rightarrow 4$ );
- до меньшего значения (например:  $3.1 \rightarrow 3$ ;  $3.9 \rightarrow 3$ ).

**Б)** Ваша процедура принимает строковую переменную, вычисляет среднее арифметическое кодов её символов и печатает его с двумя цифрами после точки.

**В)** Напечатайте с точностью три знака после точки 20 случайных чисел в диапазоне от 0 до 10. Подсказка: для формирования дробных чисел можно делить случайное целое число на другое число, например, `Random(10000)/1000`.

**Г)** Напечатайте с тремя знаками после точки 20 случайных чисел в диапазоне от 0 до 10 так, чтобы числа следовали по возрастанию. Подсказка: сравнивайте очередное число с предыдущим.

**Д)** Программа для подсчета стоимости покупок. Для каждой покупки пользователь вводит два действительных числа: вес покупки и цену за 1 кг в рублях. Признак завершения ввода данных – нулевой вес. Программа должна напечатать общую стоимость с точностью до копейки (два знака после точки) с округлением в большую сторону. Проверьте результат на калькуляторе.

**Е)** Квадратный корень. Квадрат – это равносторонний прямоугольник, его площадь вычисляется по формуле  $S=D \cdot D$ , где  $D$  – сторона квадрата. А когда площадь  $S$  известна, и надо определить сторону  $D$ ? Тогда из  $S$  извлекают квадратный корень (обозначается символом  $\sqrt{\phantom{x}}$ ). Так, если  $S=9$ , то  $D=\sqrt{9}=3$ .

Для извлечения корня в Паскале есть функция `SQRT`. Напишите собственную функцию `MySQRT`, прибегнув к методу последовательных приближений. В грубом, нулевом приближении примем  $D_0=1$ . Последующее, более точные значения  $D$  будем вычислять по формуле

$$D_{i+1} = (D_i + S/D_i) / 2$$

Так, при  $S=9$  получим  $D_1=(1+9/1)/2= \underline{5}$ ,  $D_2=(5+9/5)/2= \underline{3.4}$  и так далее, пока абсолютная разность между двумя последовательными значениями  $D$  станет пренебрежимо мала. Функция `MySQRT` должна принять число и вычислить его корень с точностью `0.0001`. Внутри функции напечатайте промежуточные значения  $D$ . Подсказка: для  $D_i$  и  $D_{i+1}$  вам потребуются лишь две локальные переменные.

**Ж)** В тесто кладут четырех главных ингредиента: муку, сахар, яичный порошок и молоко. Все это смешивается в пропорции, заданной рецептом. Например, рецепт **100:5:7:500** означает, что на 100 граммов муки кладут 5 граммов сахара, 7 граммов яичного порошка и 500 граммов молока. У пекаря есть некоторое количество всех ингредиентов, и он хочет замесить из них максимально возможное количество теста, соблюдая рецепт. Ваша программа должна ввести:

- Рецепт – это 4 целых числа.
- Исходное количество ингредиентов – это 4 действительных числа.

Программа должна напечатать:

- Общее количество полученного теста с точностью два знака после точки.
- Остатки ингредиентов – 4 числа с точностью два знака после точки.

## Глава 34

**А)** Найдите две ошибки в следующей программе.

```
var X : TNum;

type TNum = integer;

const A = 10;

begin

    X:= A+B;

end.
```

**Б)** Напишите булеву функцию **Test** и программу для её демонстрации. Функция должна проверять, делится ли без остатка первое число на второе, например:

```
Writeln( Test(20, 4) );      { true }
Writeln( Test(21, 5) );      { false }
```

**В)** Напишите целочисленную функцию **Division** для деления первого числа на второе без применения операции **DIV**. Вот примеры вызовов:

```
Writeln( Division(20, 4) );      { 5 }
Writeln( Division (21, 5) );      { 4 }
```

Подсказка: внутри функции вычитайте второе число из первого. Предотвратите деление на ноль (как результат возвращайте ноль). Сделайте два варианта: 1) деление положительных чисел, 2) деление чисел с учетом знака.

**Г)** Пусть ваша программа распечатает все множители (кроме единицы) введенного пользователем целого положительного числа, например:

```
Введите число: 60
2  2  3  5
```

**Д)** Напишите функцию для ввода целого числа. Она принимает строку-приглашение и возвращает введенное число, например:

```
X:= GetNumber('Введите стоимость покупки=');
```



## Глава 35

**А)** Полицейская база данных некоторого государства содержит номера всех автомобилей, сгруппированные в ряд множеств. Три множества составлены по типам автомобилей: легковые, грузовые, автобусы. Шесть множеств образованы по цвету автомобилей: множество белых, черных, желтых, красных, синих и зеленых.

- Пересекается ли множество легковых автомобилей с множеством грузовых? А множество желтых автомобилей с множеством черных?
- Может ли быть непустым пересечение множества желтых автомобилей с множеством автобусов?
- Свидетель дорожно-транспортного происшествия сообщил, что с места преступления скрылся **грузовой** автомобиль **синего** цвета. Как можно «вычислить» группу подозреваемых автомобилей?
- На улице висит знак: грузовым проезд запрещен. Как определить множество автомобилей, въезд которым разрешен?

**Б)** Два государства, назовем их **А** и **В**, спорят о некой территории, – каждый считает ее своей. Нарисуйте на листочке предполагаемую «карту», заштрихуйте спорную область, а затем дайте ответы на следующие вопросы:

- Как «вычислить» спорную область государств?
- «Вычислите» бесспорную область, включая оба государства.
- Заштрихуйте область, соответствующую формуле  $G = (A - B) + (B - A)$ .
- Заштрихуйте область, соответствующую формуле  $G = A + B - A \cdot B$ . Совпадает ли она с той, что вычислена по предыдущей формуле?

## Глава 36

**А)** Найдите ошибки в следующих операторах.

```
type TNumbers = set of 1..300;
    TChars = set of char;
    TBytes = set of byte;

var c1, c2 : TChars;
    b1, b2 : TBytes;
begin
    c1:= [1..9];
    c2:= ['1'..'9'];
    c2:= c2 + '0';
    c2:= c2 + [0];
    b1:= c1;
    b2:= b1 + [1,7,3];
    Writeln(b1=b2);
    Writeln(1 in b2);
    Writeln([1] in b2);
    Writeln(b1 in b2);
end.
end.
```

**Б)** Напечатайте 20 случайных чисел в диапазоне от 1 до 50 так, чтобы каждое число встретилось в распечатке лишь по разу. Подсказка: после генерации числа функцией **Random** проверьте его на вхождение в множество уже напечатанных чисел.

**В)** Введите программу решения директорской задачи (см. предыдущую страницу), а затем запустите её в пошаговом режиме (клавишей «F7»). Перед запуском вставьте переменные **S1**, **S2**, **S3**, **R** в окно обзора переменных «Watch» и

## Глава 37

**А)** Напишите процедуры для ввода и вывода множества символов. Может ли здесь счетчиком цикла быть символьная переменная?

**Б)** Напишите функцию, принимающую числовое множество и возвращающую количество содержащихся в нем элементов.

**В)** На основе первого варианта директорской программы придумайте способ поиска учеников, записавшихся более чем в один кружок. Или слабо?

**Г)** Напишите две функции, принимающие строку и возвращающие:

- строку, в которой символы исходной строки встречаются лишь по разу и следуют в алфавитном порядке, например «PASCAL» → «ACLPS»;
- то же, но порядок следования символов такой же, как в исходной строке, например «PASCAL» → «PASCL».

## Глава 38

**А)** Напишите программу для решения директорских задач и повторите подвиг контрразведчика. Или слабо?

**Б)** В небольшом островном государстве действовали забавные законы по части транспортных средств – автобусов, грузовиков и легковушек. Во-первых, общее количество автомобилей на острове не должно было превышать 256. Автомобилям назначались номера с 0 до 255, при этом соблюдались следующие правила.

Номера, делившиеся без остатка на 7, назначались автобусам. Те, что делились без остатка на 5, назначались грузовикам, а все прочие – легковушкам. Так, например, номера 35 и 70 (делятся и на 7, и на 5) доставались автобусам, а не грузовикам.

Схожие правила применялись и к окраске автомобилей, а именно.

Если номер авто делился на 4, его красили красным, если на 3 – желтым, если на 2 – белым, а остальные автомобили красили черным.

- Сформируйте три множества по классам автомобилей – автобусы, грузовики и легковушки. Вычислите количество машин каждого класса (Ответ: 37, 44, 175).
- Сформируйте четыре множества по цвету автомобилей – красные, желтые, белые и черные. Определите количество машин каждого цвета (Ответ: 64, 64, 43, 85).
- Столица того государства – деревня Кокосовка – страдала от пробок. Для борьбы с ними ввели ограничение на въезд транспорта. Так, в один из дней недели в столицу пускали только красные легковушки, белые грузовики и все автобусы. Найдите номера всех этих машин. Сколько всего автомобилей могло въехать в столицу в тот день?

**В)** Полицейская база островного государства содержала номера угнанных автомобилей – числа от 1 до 255. Это был тестовый файл такого, например, вида:

|              |
|--------------|
| 120 31 16 25 |
|--------------|

То есть, номера перечислялись через пробел и следовали в произвольном порядке, что неудобно при поиске «вручную». Ваша программа должна создать файл с номерами, упорядоченными по возрастанию. Подсказка: примените множество чисел.

**Г)** Генерация пароля длиной не менее восьми символов. В пароль входят символы трёх сортов: цифры, маленькие и большие латинские буквы, например: «**7UpJ7rst**», «**PasCa1701**». Сделайте четыре варианта так, чтобы соблюдались следующие условия:

- символ каждого сорта входит в пароль не менее двух раз, некоторые символы могут повторяться;
- все символы пароля уникальны (примените множество);
- символы одного сорта не соседствуют, например: «**Pa7sCaL5**», уникальность символов не требуется;
- символы одного сорта не соседствуют и все символы уникальны.

**Д)** Напишите булевы функции, проверяющие, является ли введенная пользователем строка правильно сформированным паролем согласно условиям предыдущей задачи.

## Глава 39

**A)** Массив **A** объявлен следующим образом.

```
var  A : array ['a'..'z'] of integer;  
     C: char;
```

Допустимо ли такое объявление и почему? Сколько элементов содержит этот массив? Какие из указанных ниже операторов будут (или могут) вызывать ошибки нарушения диапазонов?

```
A['s'] := 10;  
A['R'] := 10;  
C := 'd';    A[C] := 10;  
Readln(C);  A[C] := 10;
```

Проверьте свои решения на практике.

## Глава 40

**А)** Напишите программу для подсчета различных цифр во входном файле полицейской базы данных (считать надо именно цифры, а не числа!).

**Б)** Объявите массив из сотни целых чисел, заполните его случайными числами в диапазоне от 0 до 255 и распечатайте этот массив.

**В)** Найдите в массиве (задание Б) все элементы, хранящие число 7 (если таковые найдутся). Напечатайте индексы элементов, которые содержат это число.

**Г)** Заполните массив (задание Б) случайными числами в диапазоне от 0 до 255 так, чтобы ни одно из них не повторялось. Воспользуйтесь вспомогательным множеством чисел, где будут запоминаться сгенерированные ранее числа.

**Д)** Найдите в массиве (задание Г) наименьшее и наибольшее числа, напечатайте их, а также соответствующие им индексы элементов массива.

**Е)** Вращение массива вправо. Объявите массив из 10 чисел и заполните его случайным образом. Напишите процедуру, перемещающую 1-й элемент на 2-е место, 2-й — на 3-е место и т.д. Последний элемент должен занять 1-е место.

**Ж)** Вращение массива влево. Напишите процедуру для перемещения 2-го элемента на 1-е место, 3-го — на 2-е место и т.д. А первый элемент должен стать последним.

**И)** Напишите функцию для подсчета количества номеров в полицейской БД при условии, что одна строка может содержать несколько номеров, а некоторые строки (в т.ч. в конце файла) могут быть пустыми.

## Глава 41

**А)** Напишите программу для сортировки фамилий учеников в алфавитном порядке. Программа должна сортировать как по возрастанию, так и по убыванию фамилий (на выбор пользователя).

**Б)** Придумайте самый несправедливый способ пиратской дележки и напишите программу для неё.

**В)** Напишите программу для дележки случайным образом (как это собирались сделать пираты). Насколько отличаются результаты трех способов дележа (то есть, суммы пар слитков)?

**Г)** Напишите функцию, проверяющую, упорядочен ли числовой массив (функция должна вернуть **TRUE**, если массив упорядочен по возрастанию). Массив внутри функции передайте параметром по ссылке.



## Глава 42

**А).** Будет ли линейный поиск работать быстрее в сортированном массиве? Проверьте на практике.

**Б)** Сколько шагов двоичного поиска потребуется для массива из миллиона элементов? А из миллиарда? Сравните с числом шагов при линейном поиске.

**В)** Напишите «полицейскую» базу данных, содержащую и номера автомобилей и сведения о владельцах. Данные должны вводиться из файла, каждая строка которого содержит номер автомобиля и сведения о владельце, например:

|   |
|---|
| 123 Горбунков С.С., ул. Тепличная, д. 21, тел. 11-22-33 |
| 35 Стелькин И.Н., ул. Тенистая, д. 5, тел. 33-22-11     |

Примените массивы и учтите опыт обработки классного журнала.

**Г)** Отсортируйте массивы «полицейской» базы данных и напишите программу для двоичного поиска в ней.

**Д)** Папа Карло опасался Буратино, и прятал спички в сейфе. Код замка из четырех цифр он доверил лишь своему приятелю – честному малому Джузеппе, который не поддавался ни на какие уговоры деревянного мальчишки. Тогда тот пустился на хитрость. Ладно, – предложил Буратино, – не можешь открыть мне код, – не надо. Давай тогда в игру сыграем: я буду задавать вопросы, а ты отвечай только «да» или «нет». Первый вопрос был таким: код замка больше 5000? Через несколько минут Буратино уже рылся в папином сейфе. Сделайте программу для быстрого угадывания числа методом Буратино. Роль Буратино (угадывающего) должен исполнять компьютер.

## Глава 43

**А)** Исследуйте процедуру быстрой сортировки в пошаговом режиме (задайте небольшой размер сортируемого массива). Обратите внимание на изменение границ сортируемой части.

**Б)** Определите количество повторных входов в процедуру **QuickSort** и выходов из нее. Воспользуйтесь глобальной переменной, назовем её **Level** – «уровень». В главной программе, перед вызовом процедуры **QuickSort**, эту переменную надо обнулить, а внутри процедуры добавить следующие операторы.

В начале процедуры **QuickSort**:

```
begin
    Inc(Level); Writeln('Уровень на входе = ', Level);
```

В конце процедуры **QuickSort**:

```
    Dec(Level); Writeln('Уровень на выходе = ', Level);
end;
```

**В)** Если каждый вызов **QuickSort** делит массив примерно пополам, то наибольшее значение переменной **Level** должно составить приблизительно  $\log_2 N$  (в нашем случае размер массива **N** задан константой **CSize**). Проверьте эту догадку, скомпилировав программу для нескольких значений **CSize**.

**Г)** В одном ряду попеременно лежат дыни и арбузы. Могут ли фермеры отсортировать их за один проход ряда так, чтобы в начале оказались все дыни, а в конце ряда – все арбузы? Напишите такую программу, обозначив арбузы единицами, а дыни – нулями.

## Глава 44

**А)** Напишите процедуру, преобразующую все символы строки (латинские буквы) к верхнему регистру.

**Б)** Напишите функцию для приведения любой буквы к верхнему регистру (включая и русские). Подсказка: вспомните о таблице кодировки.

**В)** Напишите функцию для приведения любой буквы к нижнему регистру.

**Г)** Напишите собственные процедуры и функции, выполняющие то же, что и встроенные в Паскаль. Дайте своим процедурам похожие названия, например: **MyCopy**, **MyDelete** и так далее.

**Д)** Вращение строки вправо. Напишите процедуру, перемещающую 1-й символ строки на место 2-го, 2-й — на место 3-го и т.д. Последний символ должен занять 1-е место. Воспользуйтесь средствами для обработки строк.

**Е)** Вращение строки влево. Напишите процедуру для перемещения 2-го символа на место 1-го, 3-го — на место 2-го и т.д. Первый символ должен стать последним.

**Ж)** Строка содержит несколько слов (предложение). Напишите программы для решения следующих задач.

- Напечатать в столбик отдельные слова введённого предложения.
- Определить количество слов в строке.
- Равномерно расставить пробелы между словами так, чтобы длина строки стала равной 80 символам (исходная строка короче 80).

**З)** Дана строка. Напишите булеву функцию, определяющую, является ли она палиндромом (палиндром — это строка, которая читается справа налево и слева направо одинаково).

**И)** Напишите булеву функцию, определяющую, можно ли из букв первого слова составить второе (например, «клавиша» и «вилка» — **TRUE**). Учитывается только набор букв, а не их количество.

**К)** Дана строка, содержащая не менее трёх символов. Найти в ней три стоящих подряд символа, у которых сумма кодов максимальна.

**Л)** В строке найти возрастающую последовательность символов наибольшей длины (сравнивайте коды символов).

**М)** Напишите булеву функцию, проверяющую, следуют ли символы строки по неубыванию своих кодов.

**Н)** Напишите функцию для шифрования строки путём перестановки её символов, расположенных на нечётных позициях: первый символ обменивается с последним, третий – с третьим от конца и т.д.

## Глава 45

**А)** Исследуя модель танцевального кружка, можно заметить, что в любой момент одна из двух очередей обязательно пуста. В самом деле, если приходит больше мальчиков, то будет пуста «девчоночья» очередь и наоборот. Можно ли обойтись одной очередью? Придумайте, как это сделать.

Подсказка: добавьте функцию для тестирования очереди с тем, чтобы выяснить, не пуста ли она. И, если не пуста, то кто «томится» в ней первым – мальчик или девочка? Эта функция не должна изменять состояние очереди.

**Б)** Добавьте в модель сортировочной горки возможность обработки нескольких составов. Исходные «составы» (строки) должны вводиться с клавиатуры или файла, признак окончания ввода – пустая строка.

Совет: выделите действия по сортировке состава в отдельную процедуру.

**В)** Постройте механизмы очереди и стека на основе массива символов, а не строк. Какие дополнительные переменные здесь понадобятся?

## Глава 46

**А)** Постройте сверхбольшие числа на основе строковых переменных (количество цифр – не более 200).

**Б)** Напишите процедуру для вычитания сверхбольших чисел. Или слабо? Разность может быть и отрицательной!

**Г)** Жители райцентра Бюрократовка дневали и ночевали в очереди за справками. Все потому, что там применяли механический текстовый файл – огромную скрипучую книгу, которая листалась лишь в одном направлении – от начала к концу файла. Если первая буква фамилии очередного посетителя следовала по алфавиту далее, чем у предыдущего, то чиновник продолжал листать страницы с текущей позиции, а иначе открывал на первой и листал с начала. Переход от одной буквы алфавита к другой и возврат в начало занимали один час. Так, если буквы следовали в порядке «АБВ», то на выдачу справок тратилось три часа, а если в обратном порядке – «ВБА», – то шесть часов ( $3+2+1$ ). Если же первые буквы фамилий совпадали, то книгу все равно листали заново, поэтому на «БББ» тратилось шесть часов. Создайте функцию, принимающую «очередь посетителей» – строку из больших латинских букв – и возвращающую время, необходимое для выдачи всех справок.

**Д)** Томясь в бюрократической очереди, свинопас Гришка нашел способ ускорить выдачу справок путем частичного упорядочения очереди (см. задачу Г). Создайте функцию, возвращающую такую частично упорядоченную строку (воспользуйтесь множеством символов). Напишите программу для сравнения времен по условиям задач Г и Д.

## Глава 47

**А)** Напишите функцию для преобразования числа из одной системы счисления в другую. Функция должна принимать три параметра:

- строку в исходной системе счисления;
- основание исходной системы;
- основание конечной системы счисления.

Воспользуйтесь вызовами готовых функций **ConvertToNumber** и **ConvertFromNumber**.

**Б)** Программист Ник наловчился запоминать сумму цифр в номерах всех автомобилей, попадавших ему на глаза. Однажды он стал свидетелем происшествия, виновник которого скрылся. Ник сообщил полицейским только сумму цифр в номере нарушителя (сам номер Ник не помнил). Помогите полиции, и напишите программу, распечатывающую все трехзначные номера (от 1 до 999), сумма цифр которых равна **N** (значение **N** вводит пользователь).

**В)** Напишите функцию для представления чисел словами. Например, число 45 должно быть преобразовано в строку «сорок пять». Решайте задачу постепенно: сначала для однозначных и двузначных чисел, затем для более крупных. Или слабо?

**Г)** В романе «Евгений Онегин» есть такие строки: «Все предрассудки истребя, мы почитаем всех нулями, а единицами – себя». О какой системе счисления упомянул Александр Сергеевич?

**Д)** В функцию передаются три параметра: 1) число, 2) основание системы счисления, 3) символ цифры. Функция должна вернуть количество вхождений этой цифры в представление числа для указанной системы счисления.

**Е)** Напечатать все трехзначные числа, цифры которых (в десятичном представлении) различны, например: 123, 702.

**Ж)** Найти все шестизначные счастливые билеты. Счастливыми называют билеты, у которых сумма первых 3-х цифр равна сумме следующих 3-х. Например: 123 411. Напишите булеву функцию, определяющую «счастьность» билета.

**З)** В заморской стране обращались денежные купюры достоинством в 1, 2, 5, 10 и 25 пиастров. Напишите программу для кассового аппарата, определяющую наименьший набор купюр, необходимый для выдачи сдачи на указанную сумму. Например, для сдачи 33 пиастров программа напечатает: 25 + 5 + 2 + 1.

**И)** Программа шифрования текстового файла заменяет каждый символ двумя шестнадцатеричными цифрами его кода. Например, три символа '405' заменяются

на шесть символов ‘**343035**’. Символы разбивки строк не затрагиваются. Напишите программу для зашифровки и расшифровки файла по этой системе.

**К)** Чтобы усилить шифр предыдущей задачи, выполните вращение преобразованной строки на несколько позиций: влево – при зашифровке, и вправо – при расшифровке (смотрите задачи к главе 44).

**Л)** Напечатайте все числа, не превышающие 1000, такие, что делятся без остатка на каждую из своих цифр. Например: 24, 36, 184, 612. Определите количество таких чисел.



## Глава 48

**А)** Напишите программу для исследования операций сдвига (подобную программе «Р\_48\_1»).

**Б)** Наряду со сдвигами регистра, в процессорах заложены операции **циклического** (кругового) сдвига. При таком сдвиге (рис. 3) выдвигаемый бит не теряется, а попадает соответственно в младший бит (при сдвиге влево) или в старший бит (при сдвиге вправо).

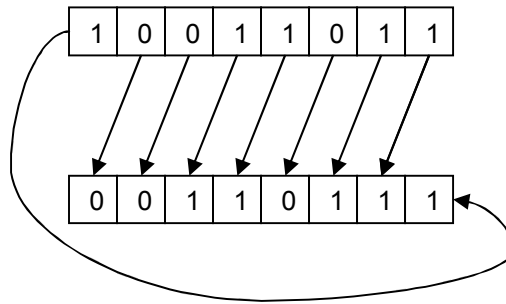


Рис. 3 – Циклический сдвиг влево

В Паскале операции циклического сдвига не предусмотрены. Напишите функции для циклического сдвига слова влево и вправо. Подсказка: перед сдвигом каждого бита надо проверить состояние теряемого бита, а затем восстановить его в младшем или старшем разряде.

## Глава 49

**А)** По ходу строительства «империи» её бывшие границы – каналы – оказываются внутри новой страны и мешают перемещению граждан, – их лучше сровнять. Дополните программу «P\_49\_1» с тем, чтобы она печатала эти бывшие границы. Или слабо?

**Б)** Измените внутреннее представление «рекламного щита» так, чтобы вместо булевых элементов использовать символы. Сделайте все необходимые изменения в программе и проверьте её работу.

**В)** В 38-й главе для нахождения простых чисел мы воспользовались множеством. К сожалению, мощность множеств в Паскале невелика (256), поэтому находить большие простые числа мы не могли. Но выход есть – это массив булевых переменных. По сути, это множество, судите сами. Объявим массив из 1000 элементов.

```
const CSize = 1000;
type TBoolSet = array [1..CSize] of Boolean;
var BS : TBoolSet;
```

Теперь условимся, что массив, заполненный значением **FALSE**, – это пустое множество. А если множество содержит числа **A** и **B**, то соответствующие им элементы массива **BS[A]** и **BS[B]** равны **TRUE**. Тогда операции с этим придуманным нами типом-множеством можно выполнять так (справа показаны аналогичные операции с обычным множеством чисел **S**).

|   |  |
|---|--|
| <code>FillChar(BS, SizeOf(BS), false);</code> | <code>{ S:= [] - пустое множество }</code>               |
| <code>FillChar(BS, SizeOf(BS), true);</code>  | <code>{ S:= [1..1000] - полное множество }</code>        |
| <code>BS[N]:= true;</code>                    | <code>{ S:= S + [N] - добавление элемента }</code>       |
| <code>BS[N]:= false;</code>                   | <code>{ S:= S - [N] - удаление элемента }</code>         |
| <code>if BS[N] then ...</code>                | <code>{ if N in S then ... - проверка вхождения }</code> |

Воспользуйтесь этим новым типом данных для поиска больших простых чисел.

**Г)** Садовая ограда. Вернувшись с курорта, фермер Лефт обнаружил на своем поле чудесно выросший сад. Для сохранения деревьев он обнес его прямоугольной оградой. Пусть ширина и высота поля заданы константами **CX** и **CY**, пустые места обозначены точками, а деревья – звездочками. Засадите поле случайным образом и распечатайте его. Затем найдите левый верхний и правый нижний углы ограды и постройте её символом решетки (ограда должна охватывать деревья, но не выходить за пределы поля). Распечатайте сад с оградой.

## Глава 50

**А)** Дополните программу «Р\_50\_2» с тем, чтобы обработать все поля записи.

**Б)** Предложите структуру записи для полицейской базы данных. Какие данные следует, по вашему мнению, включить в неё?

**В)** Напишите программу для работы с полицейской базой данных с применением записей.

**Г)** В текстовом файле тремя колонками представлены сведения о школьниках: фамилия, рост и вес. Ваша программа должна преобразовать его в три других файла, где эти же сведения отсортированы соответственно 1) по фамилиям, 2) по росту и 3) по весу учеников.

**Д)** Домино. В этой игре используют 28 костяшек, каждая из которых содержит пару чисел от 0 до 6. Например: 0:0, 1:5, 6:6. Представьте костяшку записью, а игровой набор – массивом этих записей. Заполните массив костяшек и распечатайте его. «Смешайте» костяшки случайным образом и вновь распечатайте массив. Для удобства направьте распечатку в текстовый файл.

**Е)** Карты. Колода содержит 36 карт четырех мастей: трефы и пики — черные, а бубны и червы — красные. Относительная сила карты определяется числом от 6 до 14. Представьте карту записью, содержащей её масть, цвет и силу. Представьте колоду массивом записей, сформируйте полную колоду и распечатайте в текстовый файл. «Перетасуйте» колоду и вновь распечатайте в файл. При распечатке силу карт от 11 до 14 напечатайте их названиями: валет, дама, король, туз.

## Глава 51

**А)** Какие ошибки найдет компилятор в следующей программе? Объясните их.

```
var   P1 : ^Integer;    P2 : ^String;
      N : Integer;      S : String;
begin
    P1 := @S;
    P2 := @N;
end.
```

**Б)** Будет ли работать следующая программа? В чём ошибки?

```
var   P1 : ^Integer;
begin
    P1 := 0;
    P1^ := 30;
    P1 := nil; Writeln(P1^);
end.
```

**В)** Откройте программу «P\_51\_1» и введите в окно обзора переменные **P1** и **P1^** (это делается комбинацией клавиш «Ctrl+F7»). Выполняя программу по шагам, наблюдайте эти переменные. Сделайте то же с программой «P\_51\_2».

## Глава 52

**А)** Сравните размеры переменных и указателей на них, воспользуйтесь для этого следующей программкой. Напишите нечто подобное для переменных других типов.

```
type  pb = ^byte; pw = ^word; pe = ^extended;
var   b : byte;   w : word;   e : extended;
begin
    Writeln(SizeOf(b):5, SizeOf(pb):5);
    Writeln(SizeOf(w):5, SizeOf(pw):5);
    Writeln(SizeOf(e):5, SizeOf(pe):5);
    Readln;
end.
```

**Б)** Найдите ошибки в следующей программе и объясните их.

```
var   p1, p2 : ^integer;
begin
    p1 := 10;
    p2^:= 20;
    New(p1);
    p2:= p1;
    p1^:= 'Привет!';
    Dispose(p1);
    Dispose(p2);
end.
```

**В)** Ник обожал музыку. Но компьютерный музыкальный проигрыватель раздражал программиста, поскольку при случайном выборе мелодий повторял одни песни, напрочь «забывая» о других. Предположим, в списке 10 песен, но звучали только три из них: 3, 6, 5, 6, 3, 6, 5 и т.д.

Ник создал «справедливый» проигрыватель, который выбирал мелодии иначе. Все песни состояли в одном из двух списков: «белом» или «черном». Изначально все они были в «белом» списке, и очередная мелодия выбиралась из него случайно, а после проигрывания ставилась в конец «черного». Если в этот момент в «черном» списке состояла половина мелодий, то первая мелодия из «черного» списка возвращалась в «белый». Затем снова случайно выбиралась мелодия из «белого» списка. Так гарантировалось отсутствие повторов ранее проигранных песен в течение достаточно длительного времени. Создайте программу, генерирующую случайные числа (мелодии) в диапазоне от 1 до **N** представленным выше методом. Значение **N** не превышает 255.

**Г)** Распечатывая числовое множество, мы выводили все его элементы по одному, не заботясь об экономии бумаги или места на экране. Напишите «экономную» процедуру печати множества, которая должна учитывать подряд идущие диапазоны чисел. Вот примеры желаемой распечатки:

1,5..255

0..200,210..255

0..255

2,5,7,10..20,30..40

## Глава 53

**А)** Дополните запись **TRec** полями с адресом владельца и его телефоном. Организуйте ввод и вывод этих данных (наряду с другими). Подготовьте надлежащий текстовый файл с исходными данными и проверьте работу этой версии.

**Б)** Напишите процедуру линейного поиска номера автомобиля в несортированном массиве указателей.

**В)** Напишите процедуру быстрого двоичного поиска в сортированном массиве указателей. Дополните вывод найденных данных, включая фамилию владельца и прочее.

**Г)** «Глупый» винчестер (об «умном» узнаете в задаче 54-**Д**). Рассмотрим очень упрощенную модель винчестера, быстродействие которого в основном определяется частотой вращения диска и скоростью перемещения головки чтения-записи. Время одного оборота диска примем за единицу - **квант**. За это время головка полностью читает или записывает одну дорожку. Количество дорожек на диске – 256, а нумерация идет с нуля (0...255). Время для перемещения головки на соседнюю дорожку тоже примем равным одному кванту.

Винчестером управляет контроллер, работающий несравнимо быстрее механических узлов - диска и головки. Поэтому издержками времени на его работу пренебрежем. Через известный интервал времени контроллер просматривает входную очередь, содержащую запросы на чтение или запись дорожек. Эта очередь формируется всеми запущенными программами. Мы заменим её текстовым файлом, каждая строка которого содержит по несколько чисел в диапазоне от 0 до 255 – это номера запрашиваемых дорожек. Пустая строка говорит об отсутствии запросов в текущий момент времени. Для первой строки файла сделаем исключение, поместив там лишь одно число - период просмотра этой очереди.

Контроллер «рулит» так. Прочитав список запросов (очередную строку файла), он перемещает их в свою внутреннюю очередь и далее обрабатывает её в порядке получения запросов: смещает головку в нужную позицию и выполняет чтение-запись. По ходу этой работы он следит за таймаутом, и, по истечении оно, читает следующую порцию входной очереди (строку файла). Ваша программа должна подсчитать общее время обработки запросов для набора данных из входного файла.

## Глава 54

**А)** Напишите функцию для подсчета элементов списка; она должна принимать указатель на голову списка, а возвращать целое число.

**Б)** Начертите блок-схему вставки записи в сортированный список.

**В)** Напишите процедуру удаления первого элемента списка. Или слабо?

**Г)** Напишите процедуру сортировки уже готового списка. Подсказка: последовательно извлекайте элементы из несортированного списка и вставляйте в сортированный (потребуется две головы для двух списков).

**Д)** В задаче 53-Г была представлена модель «глупого» винчестера. «Умный» винчестер отличается организацией внутренней очереди и челночным движением головки. Она следует попеременно то от внутренней дорожки к внешней, то обратно, попутно выполняя все накопившиеся в очереди запросы. Направление движения переключается тогда, когда в текущем направлении не остается запросов, поэтому головка редко достигает крайних дорожек.

Ваша программа должна подсчитать общее время обработки запросов «умным» контроллером для набора данных из входного файла, составленного по правилам для задачи 53-Г. Создайте несколько наборов таких данных и сравните время их обработки двумя типами контроллеров.

**Подсказка.** Для организации внутренней очереди контроллера примените массив чисел (счетчиков). Каждый счетчик будет хранить количество запросов для своей дорожки. При постановке запроса в очередь счетчик наращивается, а при извлечении (обработке) уменьшается.



## Глава 55

**А)** Дополните программу средствами для подсчета:

- общего количества слов в файле;
- общего количества разных слов.

Напишите для этого нужные функции.

**Б)** Измените программу так, чтобы при распечатке списка выводились относительная частота слов в процентах от их общего количества с точностью два знака после точки.

**В)** Создайте программу для подсчета в файле слов, составленных из одних и тех же символов: цифр и букв, например: «**end**» и «**deen**», «**121**» и «**221**». Для каждой такой группы слов программа должна распечатать перечень входящих в них символов (каждый символ – по разу) и количество обнаруженных слов этой группы, например, для приведенных выше слов:

|           |
|-----------|
| 1 2 - 2   |
| d e n - 2 |

Подсказка: воспользуйтесь списком записей, включающих в себя множество символов. Для слов и чисел, составленных из одинаковых символов, эти множества совпадают.

## Глава 56

**А)** В языке **Free Pascal** есть встроенная функция по имени **MemAvail** (от **Memory** – «память», **Available** – «доступный»). Функция возвращает свободный на текущий момент объем памяти в куче.

Вставьте в процедуру **Push** и функцию **Pop** следующие операторы печати.

```
Writeln('Push :', MemAvail);
```

и

```
Writeln('Pop :', MemAvail);
```

Проследите, таким образом, за изменением объема свободной памяти в куче.

**Б)** В главе 45 было высказано предположение, что для записи в танцевальный кружок достаточно одной очереди. Докажите это, создав соответствующую программу. Чем нужно дополнить процедуры работы с очередью?

## Глава 57

**А)** В старину страны континента не поддерживали дипломатических связей. Изобразите соответствующий этой ситуации граф, подразумевая под ребрами дипломатические отношения. Кстати, такой граф без ребер называется **лесом**.

**Б)** В пору расцвета континента все страны установили между собой дипломатические отношения. Нарисуйте подходящий граф.

**В)** Во время политического кризиса соседние страны перессорились между собой и разорвали дипломатические отношения. Какие ребра графа уцелели? Нарисуйте его.

**Г)** Пусть названия стран будут представлены не буквами, а словами. Возьмите карту Европы и создайте входной файл для нескольких соседних стран, например.

Франция Испания Италия Бельгия Швейцария  
Италия Франция Швейцария Словения

и так далее, перечисляя страны-соседи и отделяя их одним или несколькими пробелами. Напишите программу для ввода и вывода такого графа. Какие изменения необходимы в структуре узла?

## Глава 58

**А)** Нарисуйте граф, содержащий не менее 20 вершин, обозначьте вершины латинскими буквами и поищите в этом графе кратчайшие пути программой «Р\_58\_2».

**Б)** Пусть узлы графа – это города, а ребра – дороги между ними. Расстояния между городами разные и они известны. Как отразить в структуре графа расстояния между городами? Предложите что-нибудь.

**В)** Пусть расстояния между городами указаны в поле **mDist** записи **TLink**.

```
TLink = record      { Тип список связей }
  mLink : PNode;    { указатель на смежный узел }
  mDist : integer; { расстояние между городами }
  mNext : PLink;    { указатель на следующую запись в списке }
end;
```

Предложите формат входного файла, содержащего в числе прочего расстояния между городами.

**Г)** Пусть выбран следующий формат входного файла с учетом расстояний между городами (приведена одна строка).

```
A C 20 E 40
```

Здесь первый символ, как и ранее, обозначает текущий узел. Затем перечисляются его соседи с указанием расстояний до них. Например, между узлами «А» и «С» 20 км, а между узлами «А» и «Е» – 40 км. Напишите процедуру для ввода графа из такого файла.

**Д)** Напишите программу для поиска кратчайшего пути с учетом расстояний между городами. Потребуется слегка изменить процедуру обхода «в ширину» так, чтобы «серый» кандидат исследовал не только «белых», но и «серых» соседей, проверяя в них поле расстояния **mDist**. Если путь к соседу через кандидата окажется короче того, что уже отмечен у «серого» соседа, то следует изменить как расстояние, так и обратную ссылку в нем. Повторная установка в очередь «серого» соседа не нужна!

**Е)** Предположим, что купцам интересны не расстояния между столицами, а размер пошлин, вносимых при пересечении границ. Эти пошлины зависят от того, какую именно границу они пересекают (то есть от пары стран). Годится ли для этого случая рассмотренная модель с разными расстояниями между городами?

**Ж)** С некоторых пор купцы учредили свой ежегодный съезд – Континентальный Купеческий Конгресс, где обсуждали свои проблемы. Каждая страна отправляла на съезд по одному делегату, а расходы на пересечение границ

(визы) оплачивались из общей кассы. Посчитайте эти расходы (1 пиастр за каждое пересечение), если известна страна проведения конгресса. Считайте, что купцы следовали на съезд кратчайшими маршрутами.

**З)** Напишите программу для определения страны, где можно провести съезд с наименьшими издержками (см. задачу **Ж**).

**И)** Решите задачи **Ж** и **З** для случая разной стоимости виз на границах.

## Глава 59

**А)** Разбейте на два модуля проект «P\_58\_1» – обход графа в ширину. Какие подпрограммы должны быть «видимы» за пределами модуля? Что поместить в секцию инициализации?

**Б)** Императорские заботы. После постройки империи (главы 57 и 58), бывшие независимые государства стали её провинциями и породили новые проблемы. Для доставки туда правительственных распоряжений император нанял гонцов. Чтобы доставка была по возможности скорой, гонцы следовали лишь в одном направлении – от центра к окраинам империи. Сколько гонцов для этого нужно? – вот первый вопрос. Сколько времени потребуется для достижения самых дальних окраин, если переход из провинции в провинцию отнимает сутки? – это второй вопрос. На окраинах гонцам нужен отдых, что это за окраины, где строить гостиницы? Определите их названия.

Подсказка: возьмите за основу программу «P\_58\_1» – обход графа в ширину – и сделайте необходимые изменения в процедуре **Expand**.

## Глава 60

**А)** Создайте отладочный вариант программы обработки классного журнала (глава 31), в котором вывод результатов будет выполняться на экран. Примените условную компиляцию.

**Б)** Контрразведка перехватила несколько зашифрованных файлов, подозревая, что это тексты написанных на Паскале вирусов. Позвали Шерлока Ивановича Холмского в надежде, что тот расшифрует их. Шерлок Иванович предположил, что шифровали методом Юлия Цезаря (вспомните главу 24). Нужен ключ! После недолгих раздумий Шерлок Иванович создал программу для подбора ключей к таким текстам. Повторите еще один «подвиг контрразведчика», или слабо? Подсказка: в таких файлах после расшифровки обязательно встречаются ключевые слова **BEGIN** и **END** – воспользуйтесь этим.

**В)** Рейтинговое голосование. По избирательному закону Иксляндии каждый избиратель голосует не за одного, а за всех кандидатов, включенных в бюллетень, расставляя их в порядке своего предпочтения. Побеждает кандидат, набравший наименьшую сумму мест (если таковых несколько, то проводят второй тур). Предположим, баллотируются четыре кандидата, а бюллетени содержат следующие предпочтения избирателей:

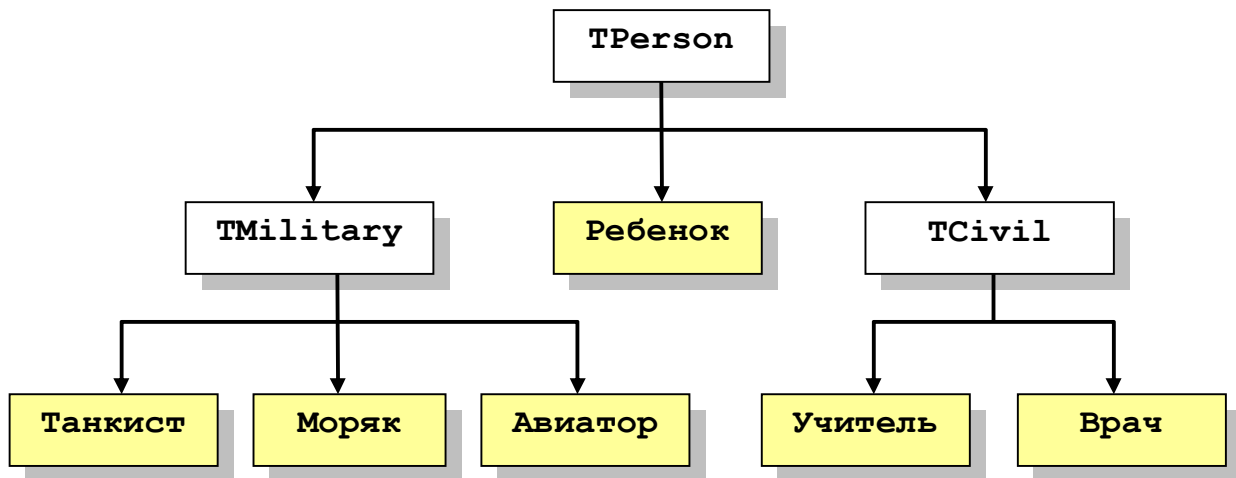
|   |   |   |   |
|---|---|---|---|
| 3 | 4 | 2 | 1 |
| 2 | 4 | 3 | 1 |
| 4 | 1 | 3 | 2 |

Здесь первый кандидат набирает сумму 10, второй – 8, третий – 7, четвертый – 5. Таким образом, побеждает четвертый кандидат в списке.

Количество кандидатов известно и равно пяти. Ваша программа принимает файл, каждая строка которого содержит 5 чисел – данные одного бюллетеня. Надо выдать список победителей голосования (одного или нескольких).

## Глава 61

**А)** Разработайте иерархию «человечьих» объектов в соответствии со следующим рисунком (новые типы объектов выделены цветом).



Дайте новым типам объектов подходящие имена, дополните их надлежащими полями, переопределите конструкторы и метод **Report**. Затем исследуйте механизм полиморфизма на предке и всех его потомках.



**Б)** Мощность Паскалевских множеств не превышает **256**, – это диапазон для байта. Часто этого бывает недостаточно. Сконструируем свой тип множеств, назовем его **TBigSet**, мощность которого составит **65536** (соответствует диапазону для типа **Word**). Оформим это множество как объект.

```

type
  TSetArray = array [0..4096] of word; { хранит 65536 бит (4096*16) }
  PSetArray = ^ TSetArray;           { тип-указатель на массив }

  TBigSet = object
    mArray : PSetArray; { указатель на динамический массив }
    Constructor Init; { создает динамический массив mArray }
    Destructor Done; { освобождает память, занятую массивом }
    procedure ClearAll; { опустошает множество }
    procedure SetAll; { делает множество полным }
    procedure Insert(N); { вставляет элемент N в множество }
    procedure Delete(N); { удаляет элемент N из множества }
    function Member(N):Boolean; { проверяет принадлежность N к множеству }
    function IsEmpty:Boolean; { проверяет пустоту множества }
    procedure Union(BS: TBigSet); { добавляет другое множество }
    procedure InterSect(BS: TBigSet); { формирует пересечение множеств }
    procedure Load(var F: text); { вводит множество из файла }
    procedure Save(var F: text); { выводит множество в текстовый файл }
  end;

```

Примените здесь сведения из главы 48, а также идеи из задачи 49-В (глава 49). Так, включение и исключение элемента **N** может быть выполнено установкой и сбросом бита в массиве **mArray**^.

```

mArray^[N div 16] := mArray^[N div 16] or (1 shl (N mod 16))
mArray^[N div 16] := mArray^[N div 16] and not (1 shl (N mod 16))

```

А объединение с другим множеством **Union(BS: TBigSet)** можно сделать так:

```

for I:=0 to 4095 do mArray^[ I ]:= mArray1^[ I ] or BS.mArray^[ I ]

```

И так далее. Напишите реализацию всех методов объекта и примените его к решетку Эратосфена и прочим задачам из главы 38.