

Г.Л.Луканкин
А.В.Диков

РЕШЕНИЕ ЗАДАЧ В СРЕДЕ ЛОГОРАЙТЕР

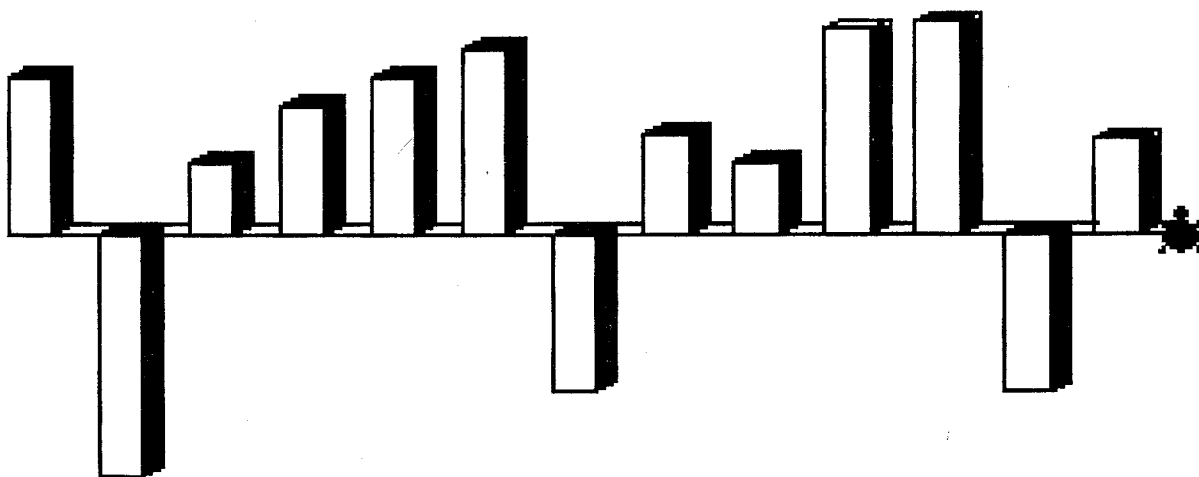
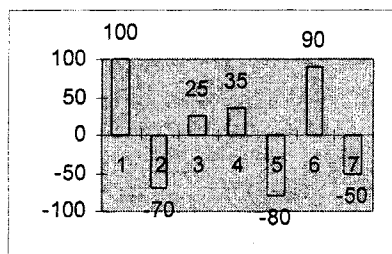
В настоящее время всё большую популярность в нашей стране приобретает язык программирования Лого. Появилась даже русифицированная версия этого языка ЛогоРайтер 2.0 фирмы LCSi. Журнал "Информатика и образование" печатал интересные статьи учителя А.Г.Юдиной, освещающие опыт её работы со школьниками на языке ЛогоРайтер. Чему же обязан своей популярностью этот язык? Ведь появился он на свет примерно в то же время, что и все известные языки программирования. Ответы на этот вопрос можно найти в книге его создателя С. Пейперта "Переворот в сознании: дети, компьютеры и плодотворные идеи". Мы лишь отметим, что, вероятнее всего, образовательный потенциал Лого уже начал проявлять себя благодаря экспериментальной работе многих учителей информатики, поэтому появилась потребность включить изучение самого Лого и его преподавания в рабочие планы педагогических вузов. Так, в Пензенском государственном педагогическом университете им. В. Г. Белинского благодаря нашим усилиям уже второй год ведётся спецкурс по данной теме.

В статье мы предлагаем разработки нескольких уроков по темам, связанным со списками и простыми словами. Каждая тема - это параграф учебника (ещё не опубликованного). Нумерация тем начинается с числа 21 потому, что учащиеся для изучения их должны освоить некоторый объём учебного материала. Предлагаемая методика выработана в результате практической работы в девятых классах лингвистической гимназии № 6 г. Пензы и теоретических размышлений авторов. Нам кажется что Лого особенно интересен для школ или классов с гуманитарным направлением обучения, так как это пока единственный язык, развивающий алгоритмическое мышление в тесном взаимодействии с образным. Кроме того, важной особенностью среды Лого является наличие кибернетического существа - черепахи, наделённой способностями, похожими на способности человека. Осознание учащимися того, что они сами выступают в роли учителей черепахи, приводит их в неописуемый восторг и делает процесс обучения желанным.

21

ЧЕРЕПАШКА ПРЕВРАЩАЕТ ЧИСЛА В СТОЛБЦЫ

Для того, чтобы можно было увидеть разницу между числами, их переводят в столбцы соответствующих длин. Рисунки из таких столбцов можно увидеть во многих газетах и журналах. Обычно набор из столбцов называют *гистограммой*, или *столбчатой диаграммой*. Научим нашу черепашку рисовать по заданному набору чисел соответствующие им столбцы.



Набор чисел определим как параметр нашей процедуры и обозначим его переменной :t. Значения переменной :t будем давать в виде списка чисел. Количество чисел может быть произвольным. Например.

№	значение :t	число элементов :t
1	[10 25 -15 2 -100 13 18]	7
2	[7 12 156 -67 -98 52]	6
3	[-15 -17 -19]	3

В языке Лого существует целый ряд служебных слов, предназначенных для работы со списками. Поэтому рассмотрим более внимательно эту структуру.

Списком называют упорядоченную последовательность элементов. Это значит, что каждый элемент списка имеет свой порядковый номер. Так, например, в приведенном выше первом списке :t элемент 13 имеет номер 6, а элемент -100 номер 5.

Кроме того, в списке выделяют голову, хвост и тело. Головой списка называется первый его элемент, хвостом - последний элемент. Все остальные элементы (кроме головы и хвоста) составляют тело списка. В первом списке :t головой является число 10, хвостом число 18, элементы 25, -15, 2, -100, 13 составляют тело данного списка.

Для работы с перечисленными выше частями списков в Лого есть следующие служебные слова - функции.

первый α	Дает первый элемент списка α
последний α	Дает последний элемент списка α
кпрв α	Дает список α без первого элемента (<u>кроме первого</u>)
кпсл α	Дает список α без последнего элемента (<u>кроме последнего</u>)
сколько α	Дает число элементов списка α

Аргументом всех перечисленных функций может быть и простое слово. Результат исполнения этих функций в данном случае будет подобным. Например, функция **первый** “Привет” будет давать значение “П”. Кроме того, на месте параметра может стоять либо выражение, значением которого должен быть список или простое слово, либо переменная с теми же значениями, либо сами эти значения.

Вернемся к задаче построения гистограммы. Чтобы черепашка строила столбец для каждого числа из списка, необходимо доставать очередное число из списка столько раз, пока не достигнем последнего или пока числа не кончатся.

повтори сколько :т [пусть “ч первый :т пусть “т кпрв :т]

Выполняя эту командную строку, компьютер присвоит переменной ч все элементы списка т по порядку, начиная с первого. Как это происходит?

№	ч	т
1	10	[25 -15 2 -100 13 18]
2	25	[-15 2 -100 13 18]
3	-15	[2 -100 13 18]
4	2	[-100 13 18]
5	-100	[13 18]
6	13	[18]
7	18	[]

Первая команда пусть (**пусть “ч первый :т**) присваивает первый элемент списка т переменной ч. Другая команда пусть (**пусть “т кпрв :т**) присваивает переменной т список т, но без первого элемента, а это означает, что список т стал на один элемент короче. Затем первая команда пусть выполняется снова, но первым элементом списка т является уже второй элемент бывшего списка т. Список из двух команд пусть будет выполняться столько раз, сколько элементов содержит исходный список т.

Для того, чтобы черепашка строила прямоугольник, ей надо дать его длину и ширину. Если мы хотим, чтобы все прямоугольники занимали поле черепахи, то ширина каждого прямоугольника вычисляется по формуле

ширина столбца = ширина экрана / число столбцов - длина промежутка

пусть "шс 300 / сколько :т - 10

Длина промежутка между столбцами - это какое-то постоянное число, например, 10 шагов черепахи. Высота столбца (длина прямоугольника) - это есть само число из списка т., однако надо учитывать, что если число больше размера экрана, то столбец не уместится на экране.

Начнем составлять процедуру.

это гистограмма :т

пусть "шс 300 / (сколько :т) - 10

пп лв 90 вп 150 пр 90 по

ось пр 90 вп 5 лв 90

повтори сколько :т

[пусть "ч первый :т

столбец :ч

пусть "т кпрв :т]

конец

это столбец :ч

вп :ч пр 90 вп :шс пр 90 вп :ч

лв 90 пп вп 10 лв 90 по

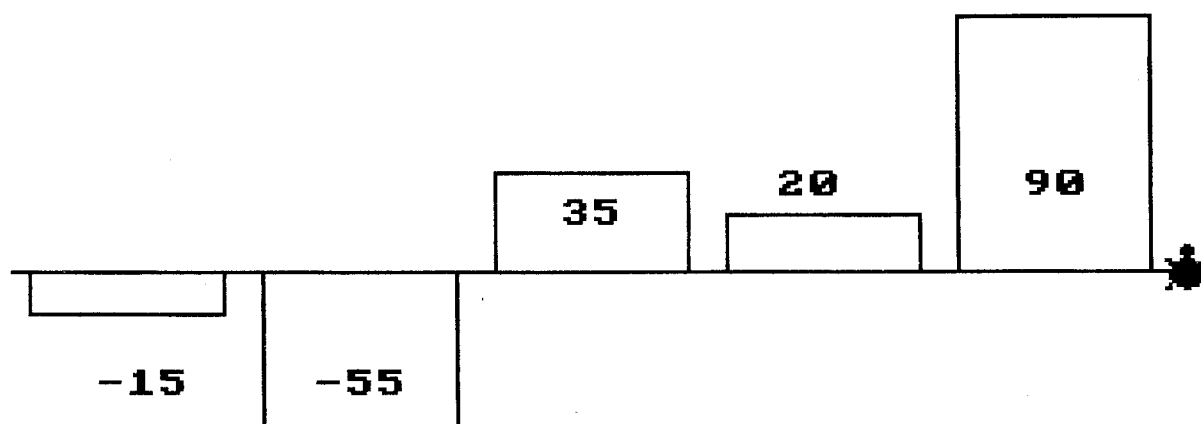
конец

это ось

пр 90 вп 300 нд 300 лв 90

конец

Команды **пп лв 90 вп 150 пр 90 по** устанавливают черепашку на позицию, с которой она начнет рисовать первый столбец. По команде **ось** черепашка проведёт горизонтальную прямую в середине экрана, от которой будут отходить столбцы вверх или вниз. Если число положительное, то столбец растёт вверх, а если отрицательное - вниз. Самое удивительное состоит в том, что при написании процедуры гистограмма мы не учитывали этот факт! Зато черепаха понимает, что если в команде вперед **:ч** число **:ч** отрицательное, то надо идти назад. По команде гистограмма **[-15 -55 35 20 90]** черепашка нарисует следующий рисунок.



X

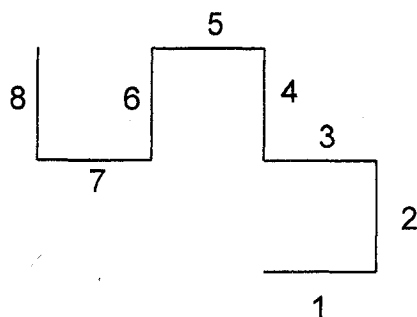
?

- а) начинаются и оканчиваются одной и той же буквой;
- б) начинаются с той же буквы, что и следующее слово;
- в) совпадают с последним словом.

22

КРИВАЯ ДРАКОНА

Научим черепашку строить кривую, которую можно получить многократным складыванием длинной полоски бумаги некоторым систематическим способом. Если после тройного складывания полосу развернуть, то получим следующую фигуру.



Кривая начинается с отрезка 1 и на каждом углу поворачивается на 90 градусов либо направо, либо налево. Построим последовательность двух чисел: 1 и 3. Эта последовательность должна отражать ход нашей кривой, то есть очередность поворотов кривой. Присвоим код 1 для поворота налево и код 3 для поворота направо. Для кривой на рисунке последовательность следующая:

1 1 3 1 1 3 3

Кривую можно продолжить дальше, и при этом окажется, что если очередной поворот имеет четный номер, то он равен тому члену последовательности, номер которого получается делением номера очередного четного поворота на 2. Например, следующий член в нашей последовательности имеет порядковый номер 8. Это число четное, значит, сам член равен члену с номером 4 ($8 / 2$), а это есть 1. Таким образом, восьмой член последовательности равен 1.

В случае, когда номер очередного поворота нечетный, член последовательности равен остатку от деления данного номера на число 4. Например, следующий неизвестный член последовательности имеет номер 9. Это нечетное число. Поэтому сам член последовательности равен 1.

Составим процедуру, которая будет создавать последовательность чисел, отражающих ход кривой, и заставим черепашку совершить путешествие по этой кривой с опущенным пером.

В качестве параметров процедуры можно взять, во-первых, число поворотов кривой (обозначим :г), во-вторых, размер отрезка (обозначим: шаг). Для того, чтобы строить в процедуре последовательность, надо завести список, в который будут добавляться очередные вычисленные значения поворотов. Сделать это помогут новые служебные слова Лого.

элемент α β	Дает элемент слова или списка β с номером α
вксп α β	Помещает значение α в <u>к</u> онец <u>с</u> писка β
внсп α β	Помещает значение α в <u>н</u> ачало <u>с</u> писка β
входит? α β	Дает истина, если α есть в β , иначе дает ложь

Во всех словах на месте параметра β должно стоять либо выражение, дающее список, либо переменная, значением которой является список, либо просто список. В функции **элемент** вместо α может быть либо число, либо числовая переменная, либо арифметическое выражение, а на месте параметра β может находиться не только то, что уже было перечислено, но и простое слово, переменная со значением простое слово, выражение со значением простое слово. В функциях **вксп** и **внсп** на месте параметра α могут находиться все возможные в Лого выражения, переменные или значения. В функции **входит?** на месте α и β могут также находиться любые выражения, переменные и значения.

Пусть у нас начальный список будет содержать 3 первых элемента последовательности. Обозначим список именем t .

пусть "т [1 1 3]

Следующий 4 элемент можно внести в конец списка командой

пусть "т вксп 1 :т

после выполнения которой будем иметь список t [1 1 3 1] уже из четырёх элементов. В процедуре построим последовательность команд, по которым будут вычисляться очередные значения поворота кривой и вноситься в конец списка t . Так как число поворотов есть параметр g , то эти команды должны выполняться g раз.

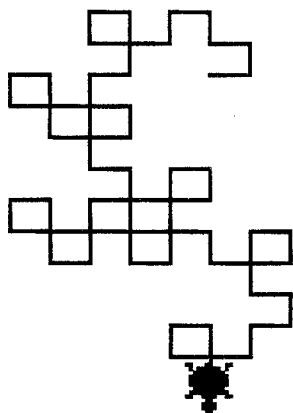


Осталось добавить команды для черепашки, чтобы она рисовала кривую. Черепашка должна делать такое количество шагов, какое мы дадим параметру шаг, затем она должна повернуть в соответствии со значением поворота, определенным в нашей последовательности поворотов.. Если значение 1, то черепаха поворачивает налево 90, если 3 - направо 90.

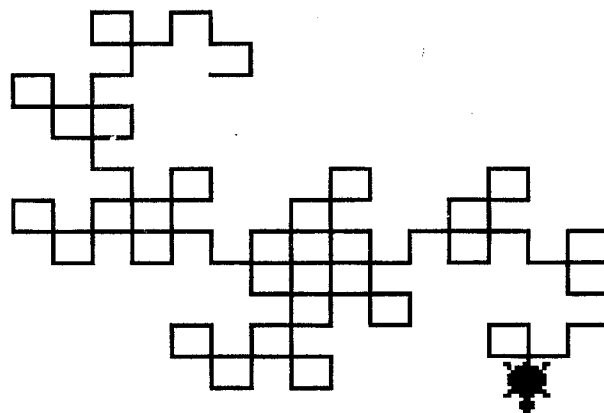
```

это дракон_1 :г :шаг
  пр 90 вп :шаг пусть "т [1 1 3] пусть "ном 4
  лв 90 вп :шаг лв 90 вп :шаг пр 90 вп :шаг
  повтори :г
    [если иначе (остаток :ном 2) = 0
      [пустить "и элемент (:ном / 2) :т] [пустить "и остаток :ном 4]
      пусть "т всп :и :т
      если :и = 1 [лв 90] если :и = 3 [пр 90] вп :шаг
      пусть "ном :ном + 1]
  конец
  
```

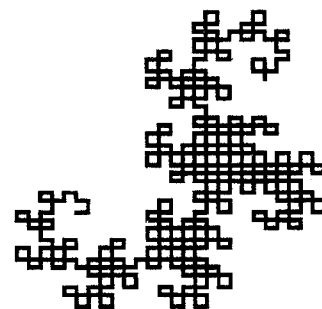
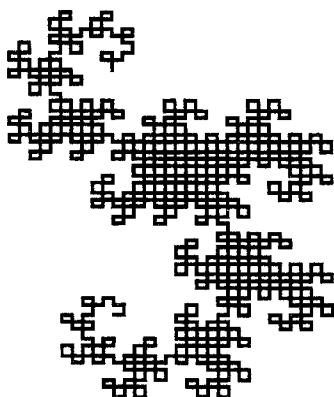
дракон_1 64 10



дракон_1 128 10



Кривые, полученные по команде дракон_1, называются кривыми дракона. Чтобы получался законченный рисунок, значение параметра г надо брать как 2 в какой-либо степени. Замечательное свойство этих кривых состоит в том, что в них отсутствуют самопересечения. Однако на рисунке это свойство неочевидно. Поэтому усовершенствуем процедуру таким образом, чтобы черепашка немного округляла прямые углы. Для этого введем дополни-

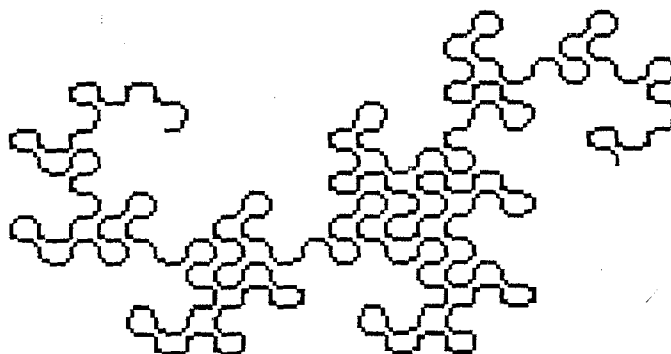


тельный параметр :к, который будет задавать число шагов для округления угла. Поворот на 90 градусов разобьем на два поворота по 45 градусов: поворот на 45, затем вперед :к и снова поворот на 45.

```

это дракон_2 :г :шаг :к
  пп новместо [-70 30] по пр 90 вп :шаг
  пусть "т [1 1 3] пусть "ном 4
  лв 45 вп :к лв 45 вп :шаг
  лв 45 вп :к лв 45 вп :шаг
  пр 45 вп :к пр 45 вп :шаг
  повтори :г
  [если иначе (остаток :ном 2) = 0
    [пусть "и элемент (:ном / 2) :т] [пусть "и остаток :ном 4]
    пусть "т вксп :и :т
    если :и = 1 [лв 45 вп :к лв 45]
    если :и = 3 [пр 45 вп :к пр 45]
    вп :шаг пусть "ном :ном + 1]
  конец
  
```

дракон_2 256 3 3



Дополнительную информацию о кривых дракона можно найти в [2, 5].

×

ВОПРОСЫ И УПРАЖНЕНИЯ

?

- 1 Какие из новых служебных слов Лого являются командами, а какие - функциями?
- 2 В каком из четырёх новых служебных слов параметром может быть простое слово?
- 3 Каким свойством обладают кривые дракона?
- 4 Напишите процедуру, которая напечатает заданный список слов в обратном порядке.

5 Составьте процедуру, которая по списку t строит два новых списка: t_1 - из положительных элементов списка t и t_2 - из остальных элементов списка t .

6 Составьте процедуру, которая по заданным двум спискам формирует третий список, состоящий из элементов:

- а) входящих в оба исходных списка;
- б) первого списка, не входящих во второй список;
- в) первого и второго списков.

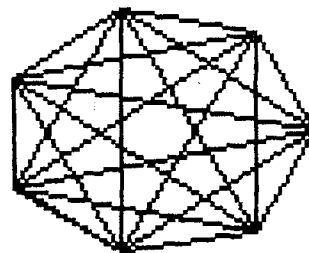
7 Составьте процедуру, которая по заданному слову определяет, является ли вторая буква слова буквой a и четвертая буква - буквой x .

8 Составьте процедуру, которая в заданном слове определяет номера:

- а) первой буквы o ;
- б) всех букв e ;
- в) всех букв p, g, t ;
- г) последней буквы a .

9 Составьте процедуру, которая по заданному списку учеников, их роста и веса, подсчитает число учеников, имеющих рост выше r и вес меньше v . Пример списка: [[Петров 50 155] [Сидоров 52 156] [Иванова 49 150]].

10 Составьте процедуру, по которой черепашка по заданному числу вершин и длине стороны будет строить правильный многоугольник, создавать список из координат вершин данного многоугольника (используйте служебное слово **место**). После того как многоугольник построен, черепашка должна из каждой вершины провести линию ко всем другим вершинам многоугольника.



11 Пилообразный текст. Составьте процедуру, по которой черепаха распечатает заданный текст в необычной форме: в первой строке - одно слово, каждая следующая строка должна быть хотя бы несколько длиннее предыдущей; когда очередная строка имеет заранее указанную максимальную длину, а текст не поместился в ней, снова повторяется то же самое, начиная со строки, состоящей из одного слова, - печатается новый зубец "пилы" [10].

Он
вступил
в темные
широкие сени,
от которых подул
холодом, как из погреба.
Из сеней он попал в комнату,
тоже темную, чуть-чуть озаренную
светом, выходящим из-под широкой
щели,
находившейся
внизу двери. Отворивши
эту дверь, он наконец очутился
в свету и был поражен представшим
беспорядком. Казалось, как будто в
доме
происходило
мытье полов и
сюда на время нагромодили
 всю мебель.

12 Составьте процедуру, дающую список чисел, полученных из заданного списка следующим образом:

$x_1, x_2, \dots, x_n$ - элементы исходного списка,

$x_1 - x_n, x_2 - x_{n-1}, \dots, x_{n-1} - x_2$ - элементы, которые надо получить.

23

ЧЕРЕПАХА НАХОДИТ ПАЛИНДРОМЫ

Палиндром (с греческого бегу назад) - "перевертень" - это слово, фраза или стих, одинаково читающиеся слева направо и справа налево. Например.

ИДИ	121
ПОТОП	202
КАБАК	1331
ИСКАТЬ ТАКСИ	42 524
ЧИН ЗВАН МЕЧЕМ НАВЗНИЧЬ	10 201

Научим нашу черепашку распознавать перевертыши. Для этого нам понадобится новое служебное слово, которое так и называется **слово**.

слово $\alpha \beta$	Дает слово, склеенное из α и β
(слово $\alpha \beta \gamma \dots$)	Дает слово, склеенное из α, β, γ и т. д.
слово? α	Дает истина, если α простое слово, иначе - ложь

Составим процедуру, в которой сначала будет формироваться слово, обратное заданному, а затем будет проверка на равенство этих двух слов. Если слова окажутся равными, то процедура будет давать слово истина, в противном случае - слово ложь.

Чтобы сформировать перевертыш, необходимо брать пустое слово и добавлять к его концу последнюю букву исходного слова, затем удалять последнюю букву исходного слова и повторять эти действия до тех пор, пока не достигнем первой буквы заданного слова.

```

это палиндром :т
  пусть "т1 "
  пусть "т2 :т
  повтори сколько :т
    [пусть "т1 слово :т1 последний :т
    пусть "т кпсл :т]
  если иначе :т2 = :т1
    [выход "истина]
    [выход "ложь]
конец

```

В процедуре палиндром есть переменная т2 для сохранения значения исходного слова, чтобы впоследствии можно было сравнить полученное и исходное слова. По команде **пиши палиндром** "казак" будет напечатано на экране слово **истина**.

т	т1
казак	"
каза	к
каз	ка
ка	каз
к	каза
"	казак

Для сравнения простых слов в процедуре палиндром используется знак равенства, но в словаре Лого имеется и специальное служебное слово равны?, которое также можно использовать вместо знака равенства.

равны? α β Дает истина, если α и β равны, и ложь в противном случае, где α β - либо простые слова или списки, либо переменные или выражения с такими значениями.

Многие задачи на обработку простых слов можно решить, используя следующие приемы:

Ф исходное простое слово "просматривается" по одному символу;

Ф формируется новое слово, исходя из условий задачи.

Например, заменить в слове все буквы т на символ +. Исходное слово обозначим переменной :ис, новое слово - переменной :нс.

это типичная_задача :ис

пусть "нс "

повтори сколько :ис

[пусть "буква первый :ис

если иначе :буква = "т [пусть "нс слово :нс "+]

[пусть "нс слово :нс :буква]

пусть "ис кпрв :ис]

выход :нс

конец

По команде **пиши типичная_задача** "тратата черепашка выведет в поле строку **+ра+а+а**.

×

ВОПРОСЫ И УПРАЖНЕНИЯ

?



1 Что такое палиндром?

2 Какие существуют приёмы для обработки простых слов?

3 Какую операцию над простыми словами совершает новое служебное слово:

а) слово;

б) (слово ...)

в) слово?

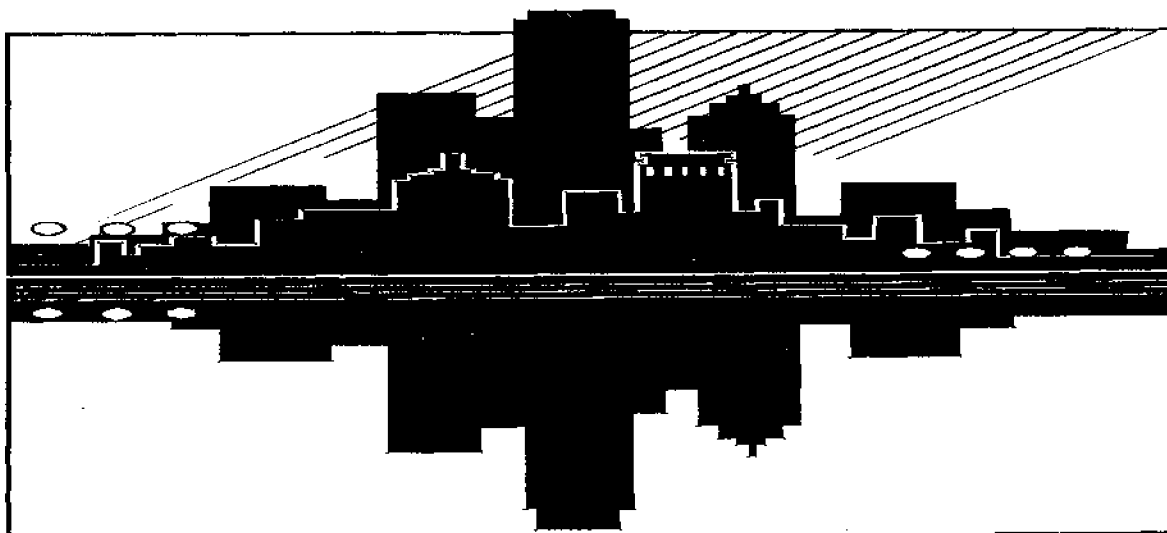
4 Напишите процедуру, которая преобразует исходное простое слово, заменив в нем:

а) все восклицательные знаки точками;

б) каждую точку многоточием (три точки);

в) каждую из групп стоящих рядом точек одной точкой.

- 5 Составьте процедуру, которая по заданному простому слову дает:
 - а) все символы, расположенные до первого двоеточия включительно;
 - б) все символы, расположенные после первого двоеточия;
 - в) все символы, расположенные между первым и вторым двоеточиями.
- 6 Составьте процедуру, которая по заданному простому слову дает:
 - а) наибольшее число идущих подряд символов /;
 - б) истину, если в слове есть пять идущих подряд букв е.
- 7 Составьте процедуру, которая в заданном простом слове исключает группы символов, расположенные между круглыми скобками. Сами скобки тоже должны быть исключены. Предполагается, что внутри пары скобок других скобок нет.
- 8 Составьте процедуру, которая в заданном простом слове заменяет группу символов *это* на группу *то*.
- 9 Составьте процедуру, которая в заданном простом слове заменяет каждый символ, встречающийся после первого вхождения символа * на -. Если нет символа *, то оставить слово без изменения.
- 10 Составьте процедуру, которая в заданном простом слове удалит все запятые, предшествующие первой точке, и заменит знаком = все цифры 4, встречающиеся после первой точки.
- 11 Составьте процедуру, которая в заданном простом слове удалит все буквы а и повторит каждый символ, отличный от а.
- 12 Составьте процедуру, которая по заданному простому слову выяснит:
 - а) входит ли в него буква ю;
 - б) верно ли, что в этом слове есть все буквы, входящие в слово *шина*;
 - в) имеется ли в этом слове пара соседствующих букв *но* или *он*.



24

ЧЕРЕПАХА ИЗУЧАЕТ ТАЙНОПИСЬ

Способ изменения письма с целью сделать текст непонятным для непосвящённых лиц называется тайнописью или криптографией. Документ или надпись, сделанные одним из способов криптографии, называются криптограммой, шифром или кодом.

М	Н	Е	Н	Е	О	Б	Х
Т	С	В	О	М	И	Д	О
Р	Е	Т	И	Т	Ь	С	Я
Ю	Ю	Ю	И	М	А	В	С

9686	9613	7546	2206
1477	1409	2225	4355
8829	0575	9991	1245
7431	9874	6951	2093
0816	2982	2514	5708
3569	3147	6622	8839
8962	8013	3919	9055
1829	9451	5781	5154

Искусство составлять секретные сообщения постигалось веками. Такие сообщения необходимы как во время войны, так и в мирное время, например, в коммерческой деятельности или деятельности спецслужб. Проблема защиты информации, хранящейся в компьютерных банках данных, может быть решена с созданием надежных шифров.

Примером простого шифра является шифр Цезаря, названный в память о том, что им пользовался еще Юлий Цезарь (100 - 44 г.г. до н.э.). Шифр основан на сдвиге букв алфавита. В алфавит Лого, как известно, входят буквы латинского и русского алфавитов и специальные знаки, всего 256 символов составляют этот алфавит.

⊠	♂	♀																																
⊠	⊡	⊢	⊣	⊤	⊥	⊦	⊧	⊨	⊩	⊪	⊫	⊬	⊭	⊮	⊯	⊰	⊱	⊲	⊳	⊴	⊵	⊶	⊷	⊸	⊹	⊺	⊻	⊼	⊽	⊾	⊿	⊠	♂	♀
⊠	⊡	⊢	⊣	⊤	⊥	⊦	⊧	⊨	⊩	⊪	⊫	⊬	⊭	⊮	⊯	⊰	⊱	⊲	⊳	⊴	⊵	⊶	⊷	⊸	⊹	⊺	⊻	⊼	⊽	⊾	⊿	⊠	♂	♀
⊠	⊡	⊢	⊣	⊤	⊥	⊦	⊧	⊨	⊩	⊪	⊫	⊬	⊭	⊮	⊯	⊰	⊱	⊲	⊳	⊴	⊵	⊶	⊷	⊸	⊹	⊺	⊻	⊼	⊽	⊾	⊿	⊠	♂	♀
⊠	⊡	⊢	⊣	⊤	⊥	⊦	⊧	⊨	⊩	⊪	⊫	⊬	⊭	⊮	⊯	⊰	⊱	⊲	⊳	⊴	⊵	⊶	⊷	⊸	⊹	⊺	⊻	⊼	⊽	⊾	⊿	⊠	♂	♀
⊠	⊡	⊢	⊣	⊤	⊥	⊦	⊧	⊨	⊩	⊪	⊫	⊬	⊭	⊮	⊯	⊰	⊱	⊲	⊳	⊴	⊵	⊶	⊷	⊸	⊹	⊺	⊻	⊼	⊽	⊾	⊿	⊠	♂	♀
⊠	⊡	⊢	⊣	⊤	⊥	⊦	⊧	⊨	⊩	⊪	⊫	⊬	⊭	⊮	⊯	⊰	⊱	⊲	⊳	⊴	⊵	⊶	⊷	⊸	⊹	⊺	⊻	⊼	⊽	⊾	⊿	⊠	♂	♀
⊠	⊡	⊢	⊣	⊤	⊥	⊦	⊧	⊨	⊩	⊪	⊫	⊬	⊭	⊮	⊯	⊰	⊱	⊲	⊳	⊴	⊵	⊶	⊷	⊸	⊹	⊺	⊻	⊼	⊽	⊾	⊿	⊠	♂	♀
⊠	⊡	⊢	⊣	⊤	⊥	⊦	⊧	⊨	⊩	⊪	⊫	⊬	⊭	⊮	⊯	⊰	⊱	⊲	⊳	⊴	⊵	⊶	⊷	⊸	⊹	⊺	⊻	⊼	⊽	⊾	⊿	⊠	♂	♀
⊠	⊡	⊢	⊣	⊤	⊥	⊦	⊧	⊨	⊩	⊪	⊫	⊬	⊭	⊮	⊯	⊰	⊱	⊲	⊳	⊴	⊵	⊶	⊷	⊸	⊹	⊺	⊻	⊼	⊽	⊾	⊿	⊠	♂	♀
⊠	⊡	⊢	⊣	⊤	⊥	⊦	⊧	⊨	⊩	⊪	⊫	⊬	⊭	⊮	⊯	⊰	⊱	⊲	⊳	⊴	⊵	⊶	⊷	⊸	⊹	⊺	⊻	⊼	⊽	⊾	⊿	⊠	♂	♀
⊠	⊡	⊢	⊣	⊤	⊥	⊦	⊧	⊨	⊩	⊪	⊫	⊬	⊭	⊮	⊯	⊰	⊱	⊲	⊳	⊴	⊵	⊶	⊷	⊸	⊹	⊺	⊻	⊼	⊽	⊾	⊿	⊠	♂	♀
⊠	⊡	⊢	⊣	⊤	⊥	⊦	⊧	⊨	⊩	⊪	⊫	⊬	⊭	⊮	⊯	⊰	⊱	⊲	⊳	⊴	⊵	⊶	⊷	⊸	⊹	⊺	⊻	⊼	⊽	⊾	⊿	⊠	♂	♀
⊠	⊡	⊢	⊣	⊤	⊥	⊦	⊧	⊨	⊩	⊪	⊫	⊬	⊭	⊮	⊯	⊰	⊱	⊲	⊳	⊴	⊵	⊶	⊷	⊸	⊹	⊺	⊻	⊼	⊽	⊾	⊿	⊠	♂	♀
⊠	⊡	⊢	⊣	⊤	⊥	⊦	⊧	⊨	⊩	⊪	⊫	⊬	⊭	⊮	⊯	⊰	⊱	⊲	⊳	⊴	⊵	⊶	⊷	⊸	⊹	⊺	⊻	⊼	⊽	⊾	⊿	⊠	♂	♀
⊠	⊡	⊢	⊣	⊤	⊥	⊦	⊧	⊨	⊩	⊪	⊫	⊬	⊭	⊮	⊯	⊰	⊱	⊲	⊳	⊴	⊵	⊶	⊷	⊸	⊹	⊺	⊻	⊼	⊽	⊾	⊿	⊠	♂	♀
⊠	⊡	⊢	⊣	⊤	⊥	⊦	⊧	⊨	⊩	⊪	⊫	⊬	⊭	⊮	⊯	⊰	⊱	⊲	⊳	⊴	⊵	⊶	⊷	⊸	⊹	⊺	⊻	⊼	⊽	⊾	⊿	⊠	♂	♀
⊠	⊡	⊢	⊣	⊤	⊥	⊦	⊧	⊨	⊩	⊪	⊫	⊬	⊭	⊮	⊯	⊰	⊱	⊲	⊳	⊴	⊵	⊶	⊷	⊸	⊹	⊺	⊻	⊼	⊽	⊾	⊿	⊠	♂	♀
⊠	⊡	⊢	⊣	⊤	⊥	⊦	⊧	⊨	⊩	⊪	⊫	⊬	⊭	⊮	⊯	⊰	⊱	⊲	⊳	⊴	⊵	⊶	⊷	⊸	⊹	⊺	⊻	⊼	⊽	⊾	⊿	⊠	♂	♀
⊠	⊡	⊢	⊣	⊤	⊥	⊦	⊧	⊨	⊩	⊪	⊫	⊬	⊭	⊮	⊯	⊰	⊱	⊲	⊳	⊴	⊵	⊶	⊷	⊸	⊹	⊺	⊻	⊼	⊽	⊾	⊿	⊠	♂	♀
⊠	⊡	⊢	⊣	⊤	⊥	⊦	⊧	⊨	⊩	⊪	⊫	⊬	⊭	⊮	⊯	⊰	⊱	⊲	⊳	⊴	⊵	⊶	⊷	⊸	⊹	⊺	⊻	⊼	⊽	⊾	⊿	⊠	♂	♀
⊠	⊡	⊢	⊣	⊤	⊥	⊦	⊧	⊨	⊩	⊪	⊫	⊬	⊭	⊮	⊯	⊰	⊱	⊲	⊳	⊴	⊵	⊶	⊷	⊸	⊹	⊺	⊻	⊼	⊽	⊾	⊿	⊠	♂	♀
⊠	⊡	⊢	⊣	⊤	⊥	⊦	⊧	⊨	⊩	⊪	⊫	⊬	⊭	⊮	⊯	⊰	⊱	⊲	⊳	⊴	⊵	⊶	⊷	⊸	⊹	⊺	⊻	⊼	⊽	⊾	⊿	⊠	♂	♀
⊠	⊡	⊢	⊣	⊤	⊥	⊦	⊧	⊨	⊩	⊪	⊫	⊬	⊭	⊮	⊯	⊰	⊱	⊲	⊳	⊴	⊵	⊶	⊷	⊸	⊹	⊺	⊻	⊼	⊽	⊾	⊿	⊠	♂	♀
⊠	⊡	⊢	⊣	⊤	⊥	⊦	⊧	⊨	⊩	⊪	⊫	⊬	⊭	⊮	⊯	⊰	⊱	⊲	⊳	⊴	⊵	⊶	⊷	⊸	⊹	⊺	⊻	⊼	⊽	⊾	⊿	⊠	♂	♀
⊠	⊡	⊢	⊣	⊤	⊥	⊦	⊧	⊨	⊩	⊪	⊫	⊬	⊭	⊮	⊯	⊰	⊱	⊲	⊳	⊴	⊵	⊶	⊷	⊸	⊹	⊺	⊻	⊼	⊽	⊾	⊿	⊠	♂	♀
⊠	⊡	⊢	⊣	⊤	⊥	⊦	⊧	⊨	⊩	⊪	⊫	⊬	⊭	⊮	⊯	⊰	⊱	⊲	⊳	⊴	⊵	⊶	⊷	⊸	⊹	⊺	⊻	⊼	⊽	⊾	⊿	⊠	♂	♀
⊠	⊡	⊢	⊣	⊤	⊥	⊦	⊧	⊨	⊩	⊪	⊫	⊬	⊭	⊮	⊯	⊰	⊱	⊲	⊳	⊴	⊵	⊶	⊷	⊸	⊹	⊺	⊻	⊼	⊽	⊾	⊿	⊠	♂	♀
⊠	⊡	⊢	⊣	⊤	⊥	⊦	⊧	⊨	⊩	⊪	⊫	⊬	⊭	⊮	⊯	⊰	⊱	⊲	⊳	⊴	⊵	⊶	⊷	⊸	⊹	⊺	⊻	⊼	⊽	⊾	⊿	⊠	♂	♀
⊠	⊡	⊢	⊣	⊤	⊥	⊦	⊧	⊨	⊩	⊪	⊫	⊬	⊭	⊮	⊯	⊰	⊱	⊲	⊳	⊴	⊵	⊶	⊷	⊸	⊹	⊺	⊻	⊼	⊽	⊾	⊿	⊠	♂	♀
⊠	⊡	⊢	⊣	⊤	⊥	⊦	⊧	⊨	⊩	⊪	⊫	⊬	⊭	⊮	⊯	⊰	⊱	⊲	⊳	⊴	⊵	⊶	⊷	⊸	⊹	⊺	⊻	⊼	⊽	⊾	⊿	⊠	♂	♀
⊠	⊡	⊢	⊣	⊤	⊥	⊦	⊧	⊨	⊩	⊪	⊫	⊬	⊭	⊮	⊯	⊰	⊱	⊲	⊳	⊴	⊵	⊶	⊷	⊸	⊹	⊺	⊻	⊼	⊽	⊾	⊿	⊠	♂	♀
⊠	⊡	⊢	⊣	⊤	⊥	⊦	⊧	⊨	⊩	⊪	⊫	⊬	⊭	⊮	⊯	⊰	⊱	⊲	⊳	⊴	⊵	⊶	⊷	⊸	⊹	⊺	⊻	⊼	⊽	⊾	⊿	⊠	♂	♀
⊠	⊡	⊢	⊣	⊤	⊥	⊦	⊧	⊨	⊩	⊪	⊫	⊬	⊭	⊮	⊯	⊰	⊱	⊲	⊳	⊴	⊵	⊶	⊷	⊸	⊹	⊺	⊻	⊼	⊽	⊾	⊿	⊠	♂	♀
⊠	⊡	⊢	⊣	⊤	⊥	⊦	⊧	⊨	⊩	⊪	⊫	⊬	⊭	⊮	⊯	⊰	⊱	⊲	⊳	⊴	⊵	⊶	⊷	⊸	⊹	⊺	⊻	⊼	⊽	⊾	⊿	⊠	♂	♀
⊠	⊡	⊢	⊣	⊤	⊥	⊦	⊧	⊨	⊩	⊪	⊫	⊬	⊭	⊮	⊯	⊰	⊱	⊲	⊳	⊴	⊵	⊶	⊷	⊸	⊹	⊺	⊻	⊼	⊽	⊾	⊿	⊠	♂	♀
⊠	⊡	⊢	⊣	⊤	⊥	⊦	⊧	⊨	⊩	⊪	⊫	⊬	⊭	⊮	⊯	⊰	⊱	⊲	⊳	⊴	⊵	⊶	⊷	⊸	⊹	⊺	⊻	⊼	⊽	⊾	⊿	⊠	♂	♀
⊠	⊡	⊢	⊣	⊤	⊥	⊦	⊧	⊨	⊩	⊪	⊫	⊬	⊭	⊮	⊯	⊰	⊱	⊲	⊳	⊴	⊵	⊶	⊷	⊸	⊹	⊺	⊻	⊼	⊽	⊾	⊿	⊠	♂	♀
⊠	⊡	⊢	⊣	⊤	⊥	⊦	⊧	⊨	⊩	⊪	⊫	⊬	⊭	⊮	⊯	⊰	⊱	⊲	⊳	⊴	⊵	⊶	⊷	⊸	⊹	⊺	⊻	⊼	⊽	⊾	⊿	⊠	♂	♀
⊠	⊡	⊢	⊣	⊤	⊥	⊦	⊧	⊨	⊩	⊪	⊫	⊬	⊭	⊮	⊯	⊰	⊱	⊲	⊳	⊴	⊵	⊶	⊷	⊸	⊹	⊺	⊻	⊼	⊽	⊾	⊿	⊠	♂	♀
⊠	⊡	⊢	⊣	⊤	⊥	⊦	⊧	⊨	⊩	⊪	⊫	⊬	⊭	⊮	⊯	⊰	⊱	⊲	⊳	⊴	⊵	⊶	⊷	⊸	⊹	⊺	⊻	⊼	⊽	⊾	⊿	⊠	♂	♀
⊠	⊡	⊢	⊣	⊤	⊥	⊦	⊧	⊨	⊩	⊪	⊫	⊬	⊭	⊮	⊯	⊰	⊱	⊲	⊳	⊴	⊵	⊶	⊷	⊸	⊹	⊺	⊻	⊼	⊽	⊾	⊿	⊠	♂	♀
⊠	⊡	⊢	⊣	⊤	⊥	⊦	⊧	⊨	⊩	⊪	⊫	⊬	⊭	⊮	⊯	⊰	⊱	⊲	⊳	⊴	⊵	⊶	⊷	⊸	⊹	⊺	⊻	⊼	⊽	⊾	⊿	⊠	♂	♀
⊠	⊡	⊢	⊣	⊤	⊥	⊦	⊧	⊨	⊩	⊪	⊫	⊬	⊭	⊮	⊯	⊰	⊱	⊲	⊳	⊴	⊵	⊶	⊷	⊸	⊹	⊺	⊻	⊼	⊽	⊾	⊿	⊠	♂	♀
⊠	⊡	⊢	⊣	⊤	⊥	⊦	⊧	⊨	⊩	⊪	⊫	⊬	⊭	⊮	⊯	⊰	⊱	⊲	⊳	⊴	⊵	⊶	⊷	⊸	⊹	⊺	⊻	⊼	⊽	⊾	⊿	⊠	♂	

СИМВОЛ α Дает символ по коду α **КОД α** Дает код символа α

По команде *пиши код* "П черепаха выведет на экран число 143, а по команде *пиши символ* 63 напечатает знак ?. Как видно из примеров, в функции *код* аргумент должен быть простым словом, но также может быть переменной или выражением со значением простое слово, а в функции *символ* - числом, либо числовой переменной, либо арифметическим выражением.

Шифр Цезаря основан на том, что каждая буква исходного сообщения сдвигается относительно алфавита вправо на какое-то заранее определенное число, называемое ключом. Сдвиг осуществляется по кругу, то есть последним символам будут соответствовать первые. Например, для ключа 3 слово ЛОГОРАЙТЕР перейдет в ОСЖСУГМХИУ.

ЛОГОРАЙТЕР
ОСЖСУГМХИУ

Составим процедуру, по которой черепаха будет делать шифровку из исходного текста. Исходный текст будем задавать в виде списка. Обозначим список переменной *:text*, ключ к шифру переменной *:key*.

Процедура *шифр_Цезаря* из списка слов *text* берет последовательно по одному слову, присваивая каждое слово переменной *newword*. Интересен новый способ извлечения слов из списка. Если

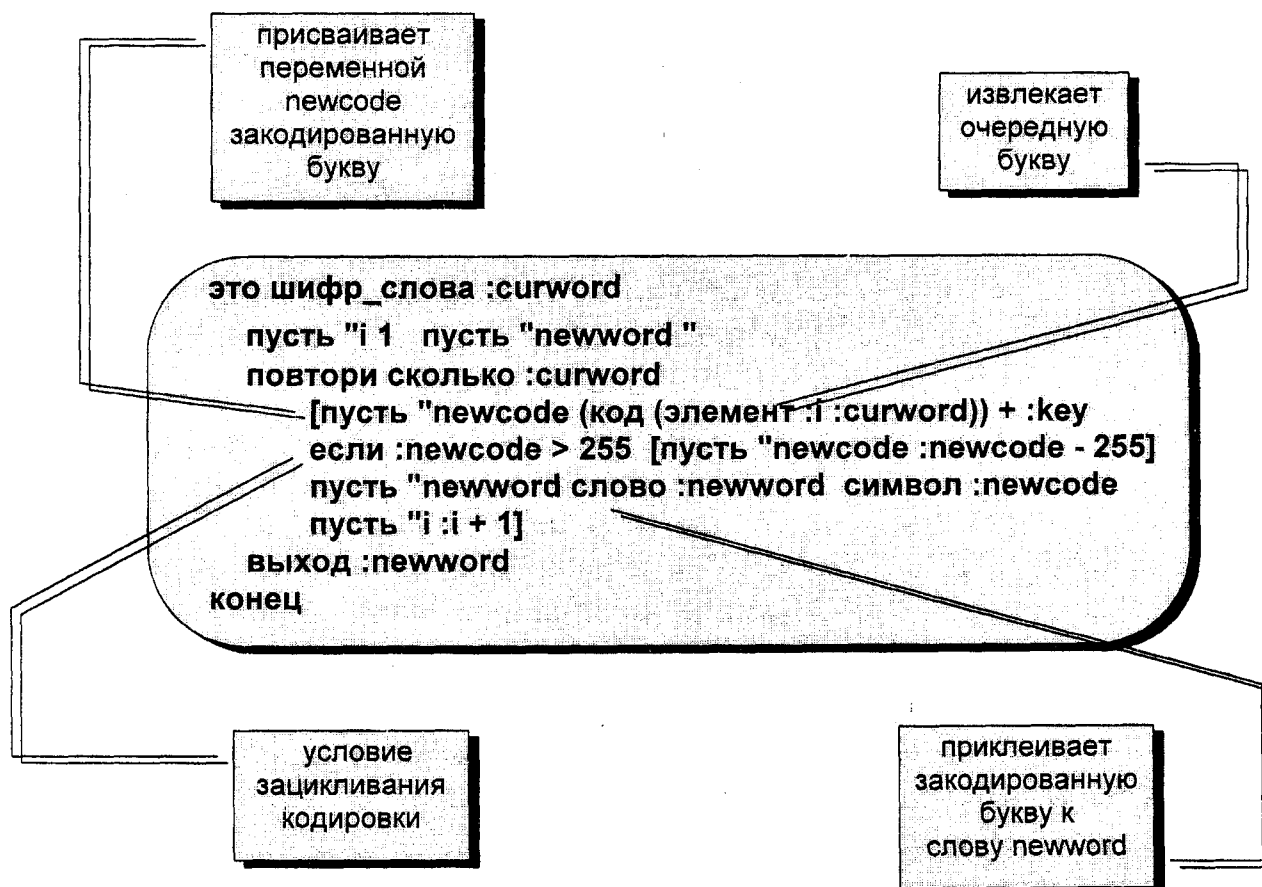
```
это шифр_Цезаря :text :key
  пусть "j" 1 пусть "newtext []"
  повтори сколько :text
    [ пусть "curword элемент :j :text
      пусть "newword шифр_слова :curword
      пусть "newtext вксп :newword :newtext
      пусть "j :j + 1 ]
  выход :newtext
конец
```

раньше мы брали из списка первый элемент, затем укорачивали список на этот элемент и повторяли всё снова, то здесь присутствует указатель номеров слов (переменная *j*), поэтому для извлечения очередного слова используется служебная функция *элемент*, одним из аргументов которой является переменная *j*.

Каждое слово, извлечённое из списка *text*, обрабатывается процедурой *шифр_слова*. Эта процедура кодирует текущее слово и создает новое, уже зашифрованное слово *newword*, которое процедурой *шифр_Цезаря* помещается в список *newtext*.

Процедура *шифр_слова* тем же способом (указатель номеров - переменная *i*) извлекает из текущего слова *curword* по одной букве, затем с помощью функции *код* находит код текущего символа, после чего к коду прибавляет ключ *key*

и к полученному результату применяет функцию *символ*, дающую уже новый, закодированный символ, который добавляется к предыдущим закодированным символам слова *newword* с помощью уже знакомого слова *слово*.



По команде

пиши шифр_Цезаря [ЛОГОРАЙТЕР - замечательный язык] 2

черепашка выведет на экран криптограмму:

НРЕРТВЛФЗТ / йвозщвфзнюпэл ёйм

Раскрыть шифр Цезаря не составляет большого труда. Например, можно для этого воспользоваться процедурой *шифр_Цезаря* и здравым смыслом. Перебирая различные значения ключа, можно натолкнуться на нужный текст. Так, например, при значении -2 (или 253) ключа *key* из приведенной выше криптограммы получим исходный текст.

Несмотря на свою простоту, шифр Цезаря может стать основой для создания нераскрываемого шифра. Усовершенствование состоит в том, что ключ для сдвига задается не для всего текста, а для каждой буквы. Кроме того, ключ выбирается случайным образом. Для расшифровки такой криптограммы требуется ключ длиной самого текста.

Дополнительную информацию по данной теме можно найти в [1, 5, 6, 10].



ВОПРОСЫ И УПРАЖНЕНИЯ



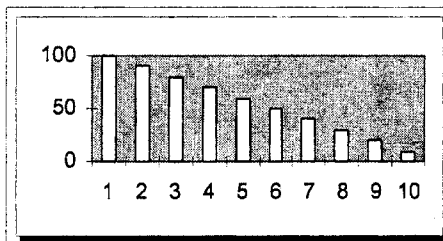
- 1 Что такое криптография? Что такое криптограмма?
- 2 Какие существуют методы кодирования текста?
- 3 Какой текст будет содержать криптограмма текста "Вундеркинд", зашифрованного методом Цезаря с ключом 255?
- 4 Какое количество символов содержится в алфавите Лого?
- 5 С помощью какого служебного слова можно узнать номер символа в алфавите? С помощью какого служебного слова можно напечатать символ по его номеру в алфавите?
- 6 Составьте процедуру, которая выведет на экран таблицу всех символов алфавита языка Лого.
- 7 Составьте процедуру, которая будет шифровать текст методом Цезаря, но ключ должен задаваться случайным образом для каждой буквы текста. Процедура должна выводить на экран криптограмму и ключ.
- 8 *Шифр Гронсфельда*. Его ключ - пятизначное число. Буквы текста разбиваются на группы по пять. Первая группа шифруется по методу Цезаря с ключом, роль которого играет первая цифра пятизначного числа, вторая группа - с ключом, равным второй цифре, и так далее. Составьте процедуру, которая будет шифровать текст методом Гронсфельда.
- 9 Составьте процедуру, которая будет шифровать последовательность из нулей и единиц (или же, например, точек и тире). Пусть $a_1, \dots, a_n$ - такая последовательность. То, что предлагается в качестве её шифра, - это последовательность $b_1, \dots, b_n$, образованная по следующему правилу:

$$b_i = a_i, \quad b_i = \begin{cases} 1, & \text{если } a_i = a_{i-1}, \\ 0 & \text{в противном случае.} \end{cases}$$
- 10 *Исправление ошибок*. Пусть по некоторому каналу связи передаётся сообщение, имеющее вид последовательности нулей и единиц (или точек и тире). Из-за помех возможен ошибочный приём некоторых сигналов: нуль может быть воспринят как единица и наоборот. Поэтому, чтобы скорректировать сообщение, при приёме каждый сигнал передается трижды, то есть он дублируется. Три последовательные цифры при расшифровке заменяются той цифрой, которая встречается среди них по крайней мере дважды. Составьте процедуру корректировки.
- 11 Составьте процедуру, которая переведёт все маленькие буквы заданного слова в большие, а большие в малые.

25

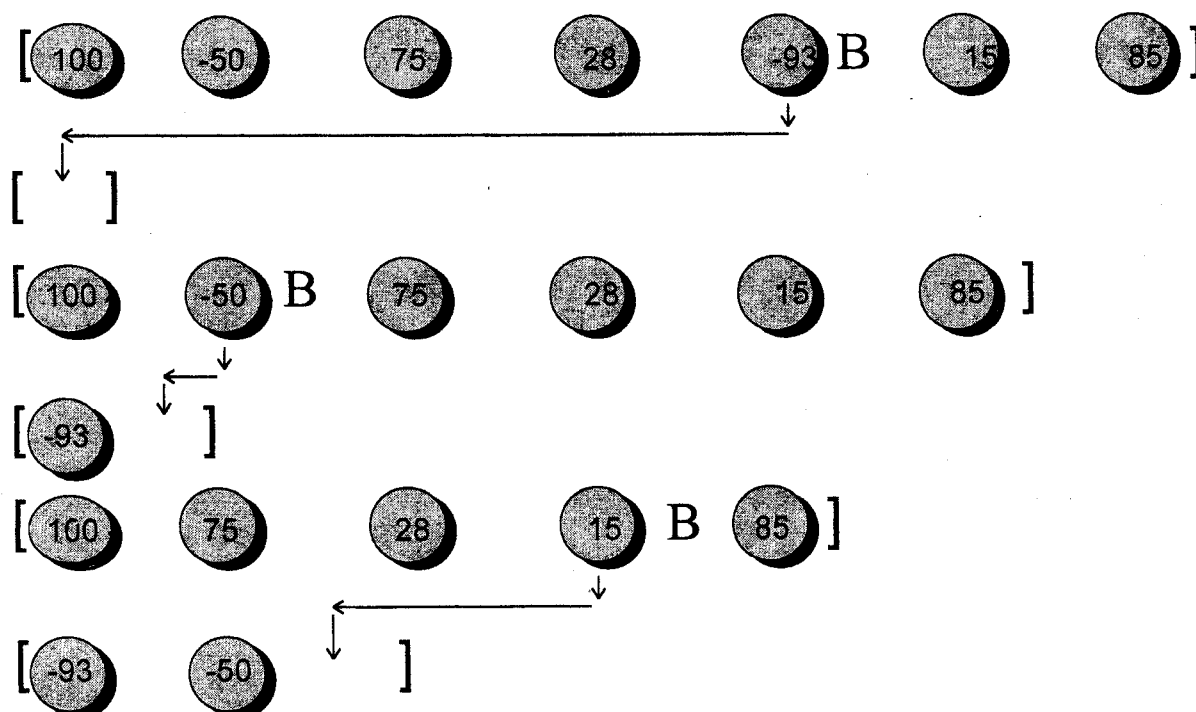
СТОЛБЦЫ ВЫСТРАИВАЮТСЯ ПО РОСТУ

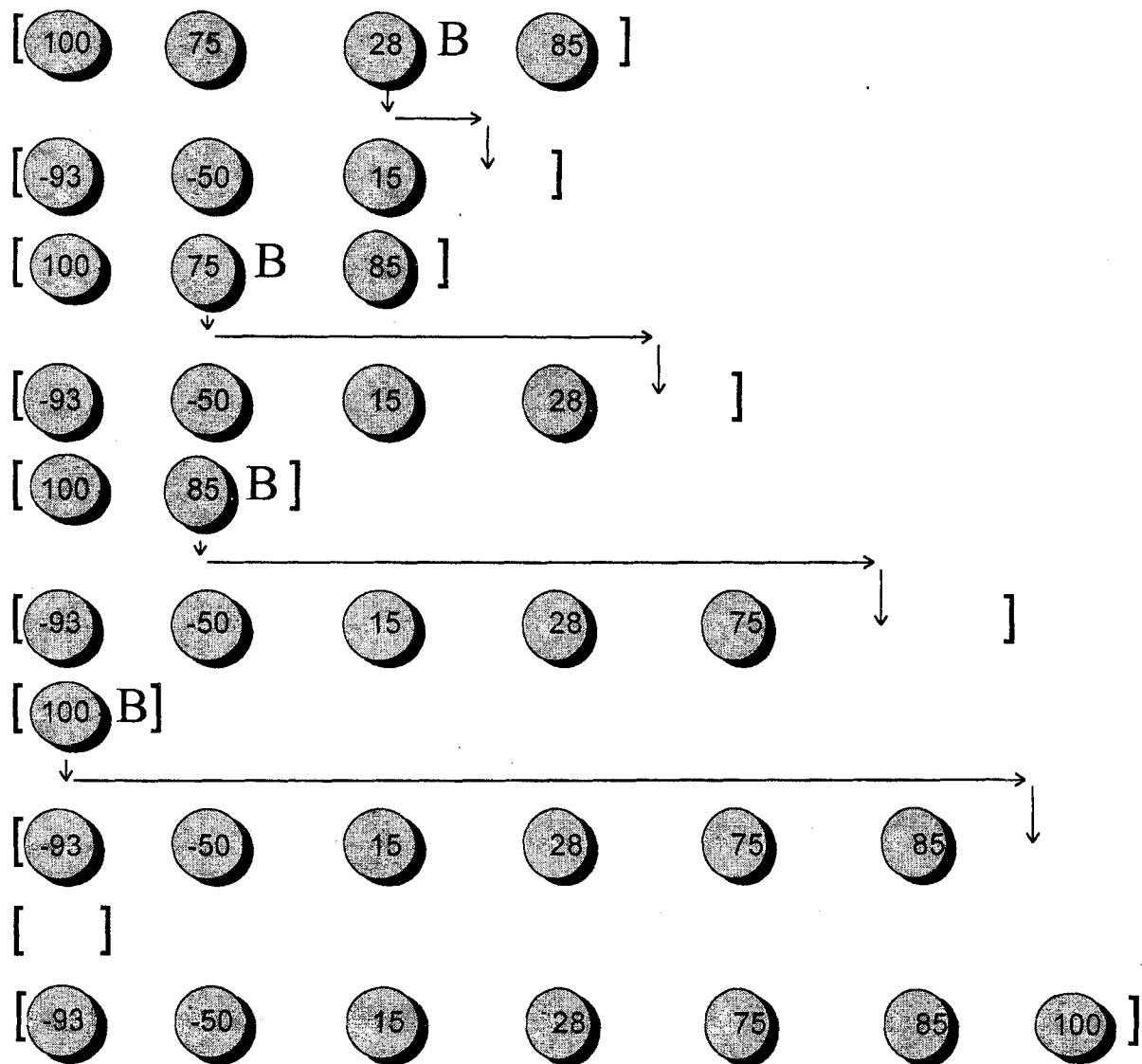
Существует много задач, в которых необходимо какие-то величины упорядочить, то есть расположить (расставить) их по убыванию или по возрастанию некоторого свойства или признака, например: расставить по росту, расположить по алфавиту. Процесс упорядочивания называют еще и сортировкой.



Имеется большое множество компьютерных программ, сортирующих данные и затем осуществляющих поиск в отсортированных данных. Дело в том, что среди упорядоченных данных можно значительно быстрее найти нужную информацию. Например, библиотечный алфавитный каталог, больничная регистратура, энциклопедические словари и т.д. Кроме того, компьютер делает работу поиска и сортировки намного быстрее человека.

Научим и мы нашу черепашку расставлять элементы списка в заданном порядке. Имеется много различных методов сортировки, среди них есть простые и сложные. Мы рассмотрим простой метод. Возьмем сначала список t , элементами которого будут только числа. Создадим список tn , в который будем помещать элементы в заданном порядке, например в порядке возрастания. Это значит, что мы должны найти в исходном списке t наименьший элемент и поместить его в конец списка tn . Затем исключить этот элемент из списка t и повторить все снова, и так будем делать до тех пор, пока в списке t не останется элементов.





Из рисунка видно, как пустой вначале список превратился в список упорядоченных по возрастанию элементов. Напишем теперь соответствующие процедуры.

```

это сортировка :т
  пусть "тн [ ]
  повтори сколько :т
    [пусть "тн всп мин :т :тн
      пусть "т исключить :т мин :т]
  выход :тн
конец

```

В процедуре сортировка есть два новых неслужебных слова, которые необходимо нам ввести в словарь языка Лого. Первое слово мин, которое должно давать наименьший элемент списка, являющегося параметром данного слова.

```

это мин :с
  пусть "i первый :с
  повтори сколько :с
    [ если :i > первый :с [ пусть "i первый :с ]
    пусть "с кпрв :с ]
  выход :i
конец

```

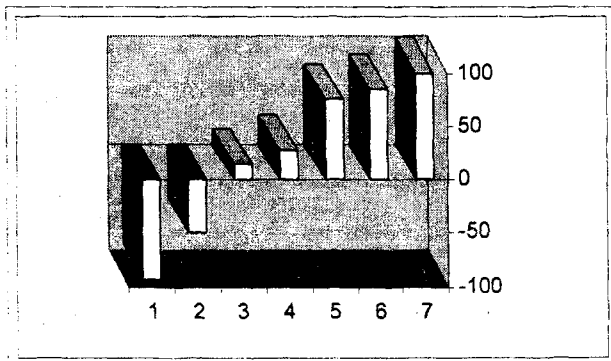
Следующее неслужебное слово - исключить. Эта функция должна возвращать список, являющийся первым ее аргументом, исключив предварительно из него элемент, являющийся вторым аргументом функции исключить.

```

это исключить :т :i
  пусть "t1 [ ]
  повтори сколько :т
    [если не (первый :т) = :i [ пусть "t1 вксп первый :т :t1 ]
    пусть "т кпрв :т ]
  выход :t1
конец

```

По команде **пиши сортировка [100 -50 75 28 -93 15 85]** черепаха выведет в своё поле уже упорядоченную последовательность чисел по возрастанию, а именно: **-93 -50 15 28 75 85 100**.



Так как новое слово сортировка является функцией, дающей список чисел, то можно использовать это слово в качестве параметра неслужебной команды гистограмма, по которой черепаха построит соответствующую упорядоченному списку столбиковую гистограмму.

гистограмма сортировка [100 -50 75 28 -93 15 85]

Однако, чтобы команда гистограмма выполнялась, необходимо присутствие в рабочей памяти компьютера соответствующей процедуры. В ЛогоРайтер есть служебные команды, которые позволяют организовать работу с процедурами, хранящимися на других листах (т.е. не на текущем листе).

возьмипроцедуры α

Загружает в память процедуры с листа α

сотрипроцедуры

Стирает в памяти внешние процедуры

процедуры

Дает список имен внешних процедур

Таким образом, имея на текущем листе процедуры, выполняющие сортировку, можно дать команду **возьмипроцедуры "GISTOGRA**, после чего коман-

да **гистограмма** будет выполняться, так как на листе GISTOGRA имеется соответствующая процедура.

Очевидно, что на месте параметра команды **возьмипроцедуры** должно стоять простое слово, являющееся именем какого-либо листа, хранящегося на диске. Процедуры с данного листа загружаются в оперативную память, но на изнанке текущего листа они не появляются.

X

ВОПРОСЫ И УПРАЖНЕНИЯ

?

1
2
3

Что такое сортировка? Для чего нужна сортировка?

Какой метод сортировки реализован в процедуре сортировка?

Как изменить процедуру сортировка, чтобы элементы выстраивались по убыванию?

4

Присутствуют ли среди новых неслужебных слов команды?

5

Появятся ли на изнанке новые процедуры после выполнения команды возьмипроцедуры?

6

Останутся ли на изнанке процедуры после выполнения команды сотрипроцедуры?

7

Составьте процедуру, дающую среднее арифметическое максимального и минимального элементов списка.

8

Задача поиска. Составьте процедуру, которая по заданному списку и элементу будет давать номер данного элемента в списке. Если же такого элемента в списке не окажется, то должно выдаваться слово нет

9

Составьте процедуру, которая по заданному списку слов находила бы самое длинное слово и выдавала его.

10

Пусть задана таблица футбольного чемпионата (например, процедурой футбол_таб), в которой все элементы, принадлежащие главной диагонали равны нулю, а каждый элемент, не принадлежащий главной диагонали, равен 2, 1 или 0 (числу очков, набранных в игре: 2 - выигрыш, 1 - ничья, 0 - проигрыш).

это футбол_таб

выход [[0 1 1 2 0 1]

[1 0 0 1 2 0]

[1 2 0 0 1 2]

[0 1 0 0 0 1]

[2 0 1 2 0 2]

[1 2 0 1 0 0]]

конец

Составьте процедуру, которая:

а) найдет число команд, имеющих больше побед, чем поражений;

б) определит номера команд, прошедших чемпионат без поражений;

в) выяснит, имеется ли хотя бы одна команда, выигравшая более половины игр.

БИБЛИОГРАФИЯ

1. Абрамов С.А. и др. Задачи по программированию. М.: Наука, 1988.
2. Аммерал Л. Программирование графики на Турбо Си. М.: Сол Систем, 1992.
3. Аммерал Л. Принципы программирования в машинной графике. М.: Сол Систем, 1992.
4. Арсак Ж. Программирование игр и головоломок. М.: Наука, 1990.
5. Болл У., Коксетер Г. Математические эссе и развлечения. М.: Мир, 1986.
6. Гарднер М. От мозаик Пенроуза к надежным шифрам. М.: Мир, 1993.
7. Гарднер М. Математические досуги. М.: Оникс, 1995.
8. Гарднер М. Математические головоломки и развлечения. М.: Оникс, 1994.
9. Гарднер М. Математические новеллы. М.: Мир, 1974.
10. Дагене В.А., Григас Г.К., Аугутис К.Ф. 100 задач по программированию. М.: Просвещение, 1993.
11. Дьяконов В.П. Язык программирования Лого. М.: Радио и связь, 1991.
12. Евстигнеев В.А. Применение теории графов в программировании. - М.: Наука, 1985.
13. Липский В. Комбинаторика для программистов. М.: Мир, 1988.
14. Лойд С. Математическая мозаика. М.: Рипол, 1995.
15. Николов Р., Сендова Е. Начала информатики. Язык Лого. М.: Наука, 1989.
16. Пайтген Х., Рихтер П. Красота фракталов. М.: Мир, 1993.