

Волкова Ризада Анверовна



ПРОГРАММИРОВАНИЕ В СРЕДЕ ЛОГО МИРЫ

Часть 6. Программирование списков

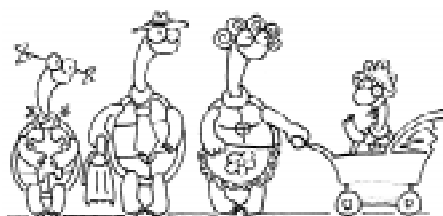
ПРОГРАММИРОВАНИЕ СПИСКОВ № 1

Задача 1-1. Добавляем в список новый элемент

Определите список членов вашей семьи. Добавьте в конец списка нового члена семьи. Результат выведите в текстовое окно.

Решение:

```
это вся_семья
пусть "семья [мама папа сын]
результат, ст пиши :семья
пусть "семья вксп "дочь :семья
результат, вставь :семья
конец
```



Задача 1-2. Считаем количество элементов в списке

Определите список, который содержит фамилии всех ваших одноклассников, присутствующих на уроке. Сравните количество элементов в этом списке с числом 16 (или любым другим числом). Если количество окажется меньше 16, выдайте сообщение «кто-то болеет». Если количество окажется равно 16, выдайте сообщение «все на уроке».

Решение:

```
это класс
пусть "присутствующие [Коля Оля Маша Тоня Ваня]
пусть "ск сколько :присутствующие
если_иначе :ск < 16 [сообщи [Кто-то болеет]]
[сообщи [все на уроке]]
конец
```

Задача 1-3. Расчет суммы элементов списка

Ознакомьтесь с программой, которая считает сумму элементов списка `сп`. Содержимое списка и значение суммы выводится в текстовое окно «результат».

Решение:

```
это v3
пусть "сп [ 1 4 6 3 7 3] ;определяется список из 6-ти чисел
пусть "сум 0 ;определяется переменная "сум - туда будет помещаться сумма
пусть "х 1 ;определяется переменная "х - счетчик элементов списка
```



```
повтори 6 [пусть "сум :сум + элемент :х :сп  пусть "х :х + 1]
результат, ст пиши :сп пиши :сум ; сам список и сумма его элементов выводится в
конец ; текстовое окно "результат".
```

Задача 1-4. Циклы и списки

Переделайте процедуру в 3-м задании так, чтобы повторение было организовано с помощью команды **всегда**, а для выхода из цикла анализировалось количество элементов в списке.

Решение:

```
это расчет_суммы2
пусть "сп [ 1 4 6 3 7 3]
пусть "сум 0
пусть "х 1
всегда [
    пусть "сум :сум + элемент :х :сп
    пусть "х :х + 1
    если :х > (сколько :сп) [автостоп]
] жди_пока [:х > (сколько :сп)]
результат, ст вставь [Элементы списка -] пиши :сп
вставь [Сумма элементов списка -] пиши :сум
конец
```

ПРОГРАММИРОВАНИЕ СПИСКОВ № 2

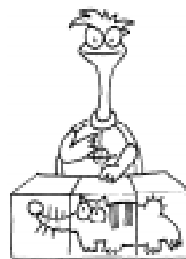
Задача 2-1. «Буриме»

Определите список **сп_вопросов**, который содержит шесть любых вопросов. Определите список **сп_ответов**, который содержит шесть ответов на эти вопросы. Программа должна выбирать случайным образом ответы и вопросы и выводить их в текстовые окна.

Подсказка: используйте команду Лого — **сл_элемент**.

Решение:

```
это задача1
пусть "сп_вопросов [
[Что вы делаете завтра?]
[Почему пропустили урок?]
[Почему опоздали?]
[О чем вы думаете?]
[Как вам пришла такая мысль?]
[Какие у вас проблемы?]
]
пусть "сп_ответов [
[Иду в кино с классом]
[Мама проспала]
[Машина у папы сломалась]
[О еде]
[Ударился головой о стену]
[Кушать хочется]
]
вопрос, ст пиши сл_элемент :сп_вопросов
ответ!, ст пиши сл_элемент :сп_ответов
конец
это подсказка1
```



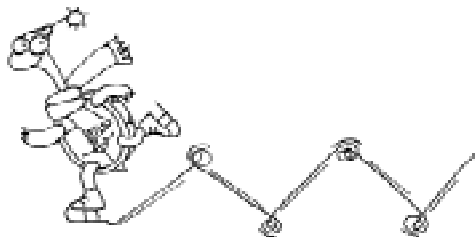
покажи [Используй команду сл_элемент :сп (выдает случайным образом выбранный элемент из списка)]
конец

Задача 2-2. Ломаная линия

Напишите программу, которая рисует ломаную линию (значения длин звеньев и углов поворота черепашки берет из списков (1 список – длины звеньев ломаной линии (от 10 до 50), 2 список – величины углов (от 0 до 180)).

Решение:

```
это задача2
  пусть "дл [15 30 5 25 40]
  пусть "уг [120 30 28 170 30]
  пусть "и 1
  по
  всегда [
  вп элемент :и :дл
  пр элемент :и :уг
  если (последний :дл) = (элемент :и :дл) [автостоп]
  пусть "и :и + 1
  ]
конец
это подсказка2
покажи [Сначала определяем два списка "длины, "углы]
покажи [Затем переменную "х, которой присваиваем значение 1]
покажи [Затем открываем цикл]
покажи [Перемещаем черепашку вперед на значение элемента №х из списка "длины]
покажи [Поворачиваем черепашку вправо на значение элемента №х из списка "углы]
покажи [Анализируем является элемент "х из списка "длины последним]
покажи [Прибавляем к переменной "х единицу и закрываем цикл]
конец
```

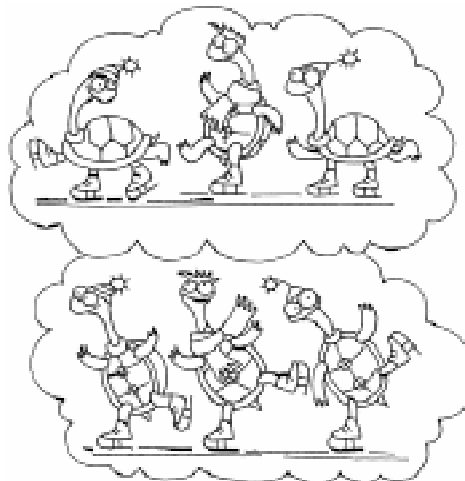


Задача 2-3. Среднее арифметическое элементов списка

Напишите процедуру, которая считает среднее арифметическое элементов списка "сп (среднее арифметическое – это сумма элементов списка, деленная на количество элементов). Содержимое списка и значение среднего арифметического выводится в текстовое окно «результат».

Решение:

```
это среднее_арифметическое
  пусть "сп [ 1 4 6 3 7 3]
  пусть "сум 0
  пусть "х 1
  повтори сколько :сп [пусть "сум :сум + элемент :х :сп
  пусть "х :х + 1]
  пусть "ср :сум / сколько :сп
  результат, ст вставь [Элементы списка - ] пиши :сп вставь [Среднее арифметическое - ] пиши :ср
конец
это подсказка3
покажи [Сначала определяем списка "сп с числами]
покажи [Затем переменную "сум, которой присваиваем значение 0]
покажи [Затем переменную "х, которой присваиваем значение 1]
покажи [Затем открываем цикл]
```



```
покажи [К переменной "sum прибавляем значение элемента №x из списка "сп]
покажи [К переменной "x прибавляем 1]
покажи [Анализируем является элемент "x из списка "сп последним. Если "Да,
                                         то останавливаем цикл (команда - стоп)]
покажи [Определяем переменную "ср, которой присваиваем значение рассчитанной суммы,
                                         делённой на количество элементов в списке]
покажи [Выводим результат в текстовое окно]
конец
```

Задача 2-4. Тест по русскому языку

Напишите процедуру, которая проверяет знание словарных слов. Ученик вставляет пропущенные буквы в слова. Напишите две процедуры.

Процедура 1 – «Получи_задание». В ней определяется два списка Один с правильно написанными словами, например – [молоко корова собака]. Другой – с пропущенными буквами [м_л_ко к_р_ва с_бака]. Второй список выводится в текстовое окно «результат».

Процедура 2 – «проверка». В ней сравнивается содержимое тестового окна (после выполнения задания) с содержимым первого списка. В случае, если значение списков совпадают, выдается сообщение «Все правильно». В противном случае выдается сообщение «Вы ошиблись».

Подсказка: Используйте команды равны?, разбери.

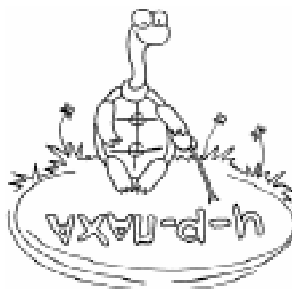
Решение:

```
это получи_задание
  пусть "сп1 [молоко корова собака]
  пусть "сп2 [м_л_ко к_р_ва с_бака]
  результат, ст вставь :сп2
конец
```

```
это проверка
  пусть "сс разбери результат
  если_иначе равны? :сп1 :сс
  [сообщи [все правильно] ]
  [сообщи [вы ошиблись]]
конец
```

```
это подсказка4
```

```
покажи [В текстовом окне результат информация находится в виде цепочки символов.]
покажи [Для преобразования символов в список используется команда разбери.
                                         (пусть "sss разбери результат)]
конец
```



ПРОГРАММИРОВАНИЕ СПИСКОВ № 3

Задача 3-1. Гласные и согласные буквы русского алфавита

Напишите процедуру, которая пишет в текстовом окне все буквы русского алфавита. Гласные буквы пишет красным цветом, согласные – синим.

Решение:

```
это задача№1
  пусть "гласные [ а е ё и о у ы э ю ]
  пусть "алф [а б в г д е ё ж з и к л м н о п р с т у ф х ц ч ш щ ь ь э ю я]
  пусть "а 1
  алфавит, ст
  всегда [
```



```

пусть "Б элемент :а :алф
если_иначе входит? :Б :гласные [нцт 15] [нцт 105]
алфавит, вставь :Б
пусть "а :а + 1
если :а > сколько :алф [автостоп]
]
конец
это подсказка1
покажи [1] определите список "глас, который содержит все гласные русского алфавита]
покажи [2] определите список "алф , который содержит все буквы русского алфавита]
покажи [3] определите переменную "х, которая будет задавать номер элемента в списке
и присвойте ей значение 1]
покажи [4] откройте цикл]
покажи [5] определите переменную "БУК, которая будет равна элементу №х из списка "алф]
покажи [6] проанализируйте входит ли эта переменна в список "глас. Если "ДА,
то определите цвет текста 15 (красный), если "НЕТ, то цвет текста 105 (синий).]
покажи [7] в тестовое окно "алфавит выведите значение переменной "БУК.
Используйте команду вставь.]
покажи [8] Не забудьте к переменной "а прибавить единицу и проанализировать
кол-во элементов в списке "алф для завершения цикла]
конец
```

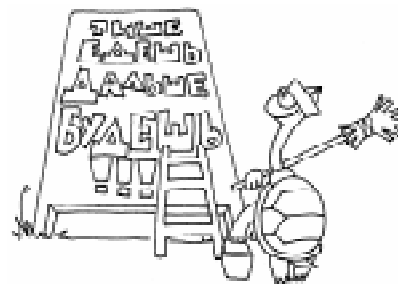
Задача 3-2. Рекламный текст

Напишите процедуру, которая запрашивает рекламный текст у пользователя и печатает его в текстовом окне «реклама». Каждое слово пишется разным цветом и увеличивающимся размером шрифта.

Решение:

```

это задача№2
спроси [введите рекламный текст]
пусть "рекл разбери ответ
пусть "и 1
пусть "рш 12
всегда [ нцт 1 + сл 200 нрш :рш
реклама, вставь элемент :и :рекл вставь символ 32
если (последний :рекл) = (элемент :и :рекл) [автостоп]
пусть "и :и + 1
пусть "рш :рш + 2
]
конец
```

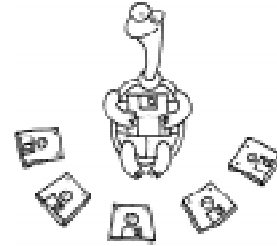


```

это подсказка2
покажи [1] Попросите пользователя ввести рекламный текст]
покажи [2] Затем определите список "рекл, в которую помещается введенный текст]
покажи [3] определите переменную "х, которая будет задавать номер элемента в списке
и присвойте ей значение 1]
покажи [4] Определите переменную "рш, которая будет задавать размер шрифта
и присвойте ей значение 12]
покажи [5] Затем откройте цикл]
покажи [6] Определите номер нового цвета текста, принимающего любое значение
от 1 до 200 (команда Лого - нцт число )]
покажи [7] Определите новый размер шрифта - значение переменной "рш
(команда Лого - нрш число )]
покажи [8] В текстовое окно реклама выводим элемент №х из списка "рекл]
покажи [9] Прибавляем к переменной "рш "2"]
покажи [10] Анализируем является элемент "х из списка "рекл последним]
покажи [11] Прибавляем к переменной "х единицу и закрываем цикл]
конец
```

Задача 3-3. Расчет суммы квадратов элементов списка

Напишите процедуру, которая считает сумму квадратов элементов списка. Элементы списка вводятся пользователем в диалоге (команда Лого – `спроси []`). Содержимое списка и значение суммы квадратов выводится в текстовое окно «результат».



Решение:

```

это сумма_квадратов
  пусть "сп [ ]
  пусть "сумкв 0
  спроси [Сколько элементов хотите ввести?]
  пусть "к_эл ответ
  повтори :к_эл [спроси [Введите элемент списка - ] пусть "сп вксп ответ :сп]
  пусть "х 1
  повтори :к_эл [  пусть "сумкв :сумкв + (элемент :х :сп) * (элемент :х :сп) пусть "х :х + 1]
  результат, ст вставь [Элементы списка - ] пиши :сп вставь [Сумма квадратов элементов
  списка - ] пиши :сумкв

```

```

конец
это подсказка3
ск
покажи [1] Сначала определяем пустой список "сп]
покажи [2] Затем определяем переменную "сумкв, которой присваиваем значение 0]
покажи [3] Затем запрашиваем количество элементов в списке]
покажи [4] Затем определяем переменную "к_эл (количество элементов),
        которой присваиваем значение введенное в диалоговом окне]
покажи [5] Затем в диалоге, по очереди вводим все элементы списка]
покажи [3] Затем переменную "х, которой присваиваем значение 1.
        Эта переменная - счетчик элементов в списке.]
покажи [4] Затем открываем цикл]
покажи [5] К переменной "сумкв прибавляем значение элемента №х из списка "сп,
        помноженное на само себя]
покажи [6] К переменной "х прибавляем 1]
покажи [Анализируем является элемент "х из списка "сп последним.
        Если "Да, то останавливаем цикл (команда - автостоп)]
покажи [7] Выводим результат в текстовое окно]
конец

```

Задача 3-4. Подсчет количества четных чисел в списке

Дан следующий список [24 25 34 33 45 67 78 87 65]. Подсчитать количество четных элементов данного списка (используйте команду `остаток`). Сам список и полученное число выведите в текстовое окно.



Решение:

```

это четные_числа
  пусть "сп [24 25 34 33 45 67 78 87 65]
  пусть "и 1
  пусть «к_чет 0
  повтори сколько :сп [
  если (остаток (элемент :и :сп) 2) = 0 [  пусть "к_чет :к_чет + 1]
  пусть "и :и + 1
  ]
  результат, ст вставь [Элементы списка - ] пиши :сп
  вставь [Четных элементов в списке - ] пиши :к_чет
конец

```

ПРОГРАММИРОВАНИЕ СПИСКОВ № 4

Задача 4-1. Перевод десятичных чисел в двоичные

Напишите процедуру, которая любое десятичное число переводит в двоичное. Десятичное число берется из текстового окна «десятичное», а двоичное выводится в текстовое окно «двоичное».

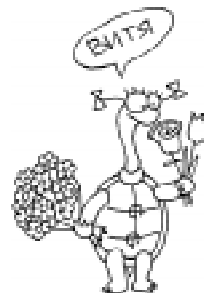
Подсказка: Обязательно сделайте контроль вводимого пользователем числа.

Решение:

```

это десять-два
двоичное, ст
спроси [Введите любое десятичное число]
пусть "ч ответ
если не число? :ч [сообщи [вводите только числа] стоп]
десятичное, ст вставь :ч
всегда [
пусть "ост остаток :ч 2
пусть "ц целое :ч / 2
двоичное, вставь :ост кнд
если :ч = 0 [автостоп]
]

```



```

конец
это подсказка1
покажи [1] Запрос десятичного числа у пользователя ]
покажи [2] определите переменную "ч, в которое поместите введенное число]
покажи [3] Проанализируйте введенные символы, если введено не число,
           то выдайте сообщение "Вводите только числа!". Остановите выполнение процедуры.]
покажи [4] откройте цикл]
покажи [5] определите переменную "ост, которая будет равна остатку деления переменной "ч
           на 2 (примитив Лого: остаток - смотри Словарь)]
покажи [6] переменной "ц присвойте значение целой части после деления её на 2
           (примитив Лого: целое - смотри Словарь)]
покажи [7] в тестовое окно "двоичное выведите значение переменной "о.
           Используйте команду вставь. ]
покажи [8] Курсор верните назад на 1 позицию]
покажи [9] проанализируйте равна ли переменная "ч 0 для завершения цикла]
конец

```

Задача 4-2. Перевод двоичных чисел в десятичные

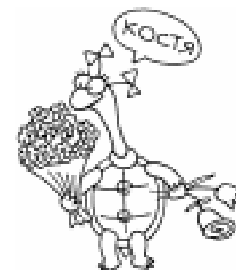
Напишите процедуру, которая двоичное число переводит в десятичное. Двоичное число берется из текстового окна «двоичное». Десятичное число выводится в текстовое окно «десятичное».

Решение:

```

это два-десять
десятичное, ст
пусть "чдв двоичное
пусть "х 0 ;счетчик шагов в цикле
пусть "чисдес 0 ;десятичное число
пусть "с сколько :чдв ;кол-во знаков в двоичном числе
повтори :с [
пусть "эл элемент (:с - :х) :чдв ;берем по очереди цифры; из "чдв с конца
если :эл > 1 [сообщи [число не двоичное] останов]
пусть "чисдес :чисдес + (степень 2 :х) * :эл
пусть "х :х + 1
]

```



```

десятичное, пиши :чисдес
конец
это подсказка2
покажи [1] Определите переменную "чдв, в которую поместите информацию
                                     из текстового окна "двоичное]
покажи [2] определите переменную "х, которая будет задавать номер элемента в списке
                                     и присвойте ей значение 0]
покажи [3] Определите переменную "чисдес, куда в итоге поместится десятичное число
                                     и присвойте ей значение 0]
покажи [4] Определите переменную "с равную количеству элементов в «чдв]
покажи [5] Затем откройте цикл (количество повторений известно,
                                     поэтому рациональнее использовать команду Лого повтори]
покажи [6] Определите переменную «эл, которая равна значению элемента № (:с - :х) из "чдв]
покажи [7] Проанализируйте: если :эл больше 1, то введенное число - не двоичное
                                     и остановите выполнение процедуры]
покажи [8] Прибавляем к переменной "чисдес значение переменной "эл,
                                     умноженное на 2 в степени :х]
покажи [9] Прибавляем к переменной "х единицу ]
конец

```

Задача 4-3. Перевод восьмеричных чисел в десятичные

Напишите процедуру, которая восьмеричное число переводит в десятичное. Восьмеричное число берется из текстового окна «восьмеричное».

Решение:

```

это восемь-десять
пусть "ч восьмеричное
десятичное, ст
пусть "х 0
пусть "чис 0
пусть "с сколько :ч
повтори :с [
пусть "эл элемент (:с - :х) :ч
если :эл > 7 [сообщи [число не восьмеричное] останов]
пусть "чис :чис + (степень 8 :х) * :эл
пусть "х :х + 1
]
это подсказка3
покажи [ смотри предыдущую задачу ]
конец

```

Задача 4-4. Перевод восьмеричных чисел в двоичные

Напишите процедуру, которая восьмеричное число переводит в двоичное. Восьмеричное число берется из текстового окна «восьмеричное». А двоичное выводится в текстовое окно «двоичное».

Решение:

```

это восемь-два
пусть "8-ич [ 0 1 2 3 4 5 6 7]
пусть "2-ич [000 001 010 011 100 101 110 111]
двоичное, ст
пусть "ч восьмеричное
пусть "х 1
повтори сколько :ч [
пусть "п элемент :х :ч
пусть "т элемент (:п + 1) :2-ич

```




```

двоичное,
если (сколько :т) = 2 [вставь "0]
если (сколько :т) = 1 [вставь "«00]
вставь :т вставь символ 32
пусть "х :х + 1
]
конец
это подсказка4
покажи [Здесь придется воспользоваться своим умом. Удачи!]
конец
    
```

ДОПОЛНИТЕЛЬНЫЕ ЗАДАЧИ ПО ТЕМЕ «ПРОГРАММИРОВАНИЕ СПИСКОВ»

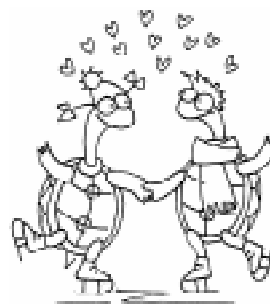
Задача 1. Объединяем два списка

Дано два списка. Первый – список имен всех мальчиков в классе. Второй – список имен всех девочек в классе. Составить третий список, содержащий имена всех учеников класса.

Решение:

```

это класс
пусть "девочки [Вика Майя Оля Сабина]
пусть "мальчики [Антон Игорь Илья Паша Рома Миша]
пусть "весь_класс предложение :список1 :список2
результат, ст вставь :весь_класс
конец
    
```



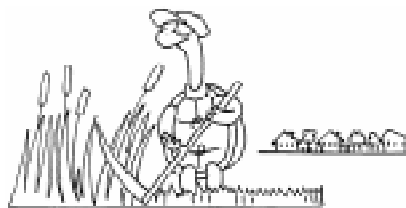
Задача 2. Косим траву

Создайте небольшую игру – по листу проекта перемещается коса (следует за указателем мыши) и скашивает ближайшие травинки (черепашки меняют форму).

Решение:

```

это коси_коса
пусть "черепашки [цв1 цв2 цв3 цв4 цв5 цв6 цв7 цв8 цв9 цв10 ]
пусть "граница 50
всегда [
    коса, нм мм
    перебор [к :черепашки ] [для :к если (путь "коса) < :граница [для :к нф 13]]
]
конец
это вернуть
перебор [к :черепашки] [для :к нф 7]
конец
    
```



Задача 3. Косим новую траву

Создайте небольшую игру – в поле вырастает трава (появляются новые черепашки, случайным образом располагаются на листе проекта), затем появляется коса (следует за указателем мыши) и скашивает ближайшие травинки (черепашки меняют форму).

Подсказка: Для получения имени новой черепашки используйте команду Лого слово.

Решение:

```

это коси_коса2
пусть "черепашки [ ] ; список черепашек
пусть "граница 50 ; расстояние от косы до травы
пусть "и 1 ; счетчик травянок
повтори 10 [
    
```



```

    пусть "ич слово "цв :и  нч :ич      ; сначала создается имя черепашки
                                           ; - склеиваются буквы цв и число от 1-го до 10-ти.
нов_х (-200 + сл 400) нов_у (-100 + сл 150) нф 7 пч ; черепашка перемещается на
                                           ; случайное место, меняет форму и показывается
    пусть "черепашки всп :ич :черепашки      ; имя черепашки заносится в список
    пусть "и :и + 1
]
всегда [
    коса, нм мм
    перебор [к :черепашки ] [для :к если (путь "коса) < :граница [для :к нф 13]]
]
конец
это вернуть2
перебор [к :черепашки] [удали :к ]
конец

```

Задача 4. Ищем ближайший цветок (поиск минимального элемента списка)

Создайте небольшую игру – пчелка летит к ближайшему цветку и садится на него

Решение:

```

это полет_пчелы
    пусть "черепашки [цв1 цв2 цв3 цв4 цв5 цв6 цв7 цв8 цв9 цв10 ]
    пусть "расст [ ] ; расстояние от пчелы до цветов
    перебор [к :черепашки ] [для :к пусть "расст всп (целое путь "пчелка) :расст ] ; заполняется
                                           ; список расстояний от пчелы до цветов
    пусть "ном_ближ_ч 1 ; номер ближайшего цветка - 1 (пока)
    пусть "ближ_р элемент 1 :расст ; за минимальное расстояние берется расстояние до первого цветка цв1.
    пусть "и 1
    повтори (сколько :расст) [
        если (элемент :и :расст) < :ближ_р [пусть "ближ_р (элемент :и :расст) пусть "ном_ближ_ч :и ]
        пусть "и :и + 1
    ]
    пчелка, курс_на (элемент :ном_ближ_ч :черепашки) плавно :ближ_р 1 ; пчела лети к ближайшему цветку
конец

```



Задача 5. Облетаем все цветы (поиск минимального элемента списка)

Создайте небольшую игру – пчелка облетает все цветы, каждый раз выбирая ближайший.

Решение:

```

это полет_пчелы2
    пусть "черепашки [цв1 цв2 цв3 цв4 цв5 цв6 цв7 цв8 цв9 цв10 ]
    пусть "посещ_цветы [ ] ; список черепашек, которые пчелка уже посетила
    повтори сколько :черепашки [
    пусть "расст [ ] ; расстояние от пчелы до цветов
    перебор [к :черепашки ] [для :к если_иначе не входит? :к :посещ_цветы
        [пусть "расст всп (целое путь "пчелка) :расст ]
        [пусть "расст всп 1000 :расст] ; если цветок уже посещался пчелкой,
                                           ; то расстояние до него приравниваем 1000,
                                           ; чтобы этот цветок гарантированно
                                           ; не оказался ближайшим
    ] ; заполняется список расстояний от пчелы до цветов

```



```

пусть "ном_ближ_ч 1 ;номер ближайшего цветка - 1 (пока)
пусть "ближ_р элемент 1 :расст ;за минимальное расстояние берется расстояние
;до первого цветка цв1.

пусть "и 1
повтори (сколько :расст) [
  если (элемент :и :расст) < :ближ_р [пустить «ближ_р (элемент :и :расст) пусть "ном_ближ_ч :и ]
  пусть "и :и + 1
]
пчелка, курс_на (элемент :ном_ближ_ч :черепашки) плавно :ближ_р 1 ;пчела лети к ближайшему цветку
пусть "посещ_цветы вксп (элемент :ном_ближ_ч :черепашки) :посещ_цветы ;пополняется список
;посещенных цветов
]
конец

```

Задача 6. Строим мальчиков по росту (сортировка элементов списка)

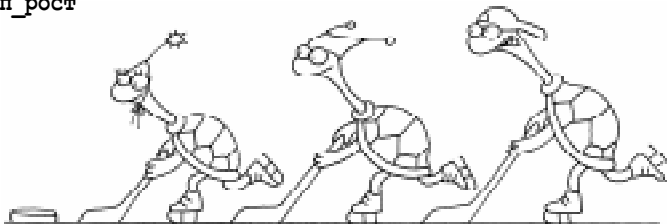
Создайте небольшую игру – на поле появляются 10 игроков разного роста, затем по команде они строятся по росту.

Решение:

```

это команда-на-поле
пусть "сп_чер [ ]
пусть "сп_рост [ ]
много_раз [к 10] [ пусть "имя_ч (слово "игрок :к ) пусть "сп_чер вксп :имя_ч :сп_чер
нч :имя_ч нов_х -200 + сл 400 нов_у -100 + сл 150 нф 41 пусть "раз_чер (20 + сл 20) нрз :раз_чер пч
  пусть "сп_рост вксп :раз_чер :сп_рост
]
конец

```



```

это стройся!
пусть "сп_чер_сорт [ ]
пусть "сп_рост_сорт [ ]
повтори сколько :сп_чер [
макси ;ищем максимальный элемент в списке ростов.
пусть "сп_рост_сорт вксп :макс :сп_рост_сорт
пусть "сп_чер_сорт вксп (элемент :№макс :сп_чер ) :сп_чер_сорт
повтори :№макс [пустить "сп_рост вксп ( элемент 1 :сп_рост) :сп_рост
;макс
  пусть "сп_рост кпсл :сп_рост
повтори :№макс [пустить "сп_чер вксп ( элемент 1 :сп_чер) :сп_чер пусть "сп_чер кпрв :сп_чер]
  пусть "сп_чер кпсл :сп_чер
]
пусть "х1 -200
перебор [к :сп_чер_сорт] [для :к нов_у -100 нов_х :х1 пусть "х1 :х1 + 35 жди 3]
конец

это макс
пусть "макс элемент 1 :сп_рост
  пусть "№макс 1
  пусть "и 1
повтори сколько :сп_рост [
если :макс < ( элемент :и :сп_рост) [ пусть "макс ( элемент :и :сп_рост) пусть "№макс :и ]
  пусть "и :и + 1
]
конец

это вернуть6
перебор [к :сп_чер_сорт] [удали :к]
конец

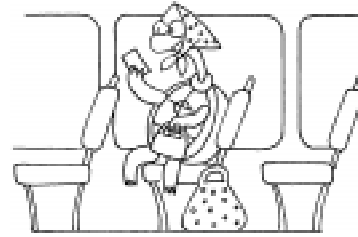
```

Задача 7. Поиск счастливых билетов

В текстовое окно вывести все четырехзначные числа, у которых сумма двух первых цифр равна сумме двух последних.

Решение:

```
это счастливый_билет
счастливые_номера, ст
  пусть "х 1000
  пусть "к 0
  повтори 9999 - 1000 [
    если (элемент 1 :х) + (элемент 2 :х) = (элемент 3 :х) + (элемент 4 :х)
      [пусть "к :к + 1 счастливые_номера, вставь :х вставь символ 32]
  ]
  пусть "х :х + 1
]
вставь "всего- вставь :к вставь "счастливых_номеров
конец
```

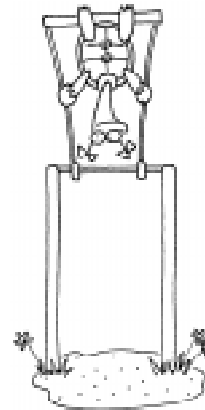


Задача 8. Слова-перевертыши

Напишите процедуру, которая любое слово, заданное пользователем, пишет наоборот (начиная с последней буквы).

Решение:

```
это перевертыш
слово!, ст !оволс, ст
спроси [напиши любое слово !]
  пусть "aaa ответ
  слово!, вставь :aaa
  пусть "х сколько :aaa
  всегда [
    !оволс, вставь элемент :х :aaa
    пусть "х :х - 1
    если :х = 0 [автостоп]
  ]
конец
```

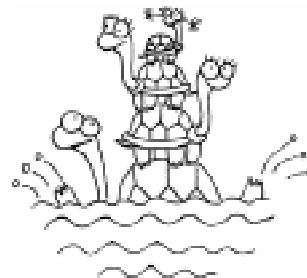


Задача 9. Строим пирамиду из слова

Построить пирамиду из слова. На первой строке пишется все слово, на второй слово без последней буквы, на третьей слово без двух последних букв и так до тех пор, пока не кончатся все буквы в слове

Решение:

```
это пирамида
пирамидка, ст
спроси [напиши любое слово !]
  пусть "ккк ответ
  пусть "х сколько :ккк
  всегда [
    пирамидка, пиши :ккк
    пусть "«ккк кпсл :ккк
    если :х = 1 [автостоп]
    пусть "х :х - 1
  ]
конец
```

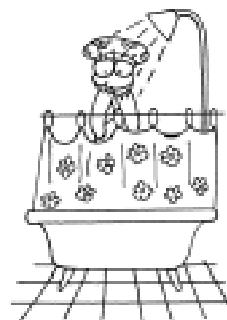


Задача 10. Да и нет не говорить ...

Запрограммировать известную детскую игру «Да и нет не говорить, черное с белым не носить».

Решение:

```
это старт!!!
путь "х 1
путь "зс [ да нет чёрный  черное черная чёрная чёрное  черный  белый белая белое]
путь "вопрос [
    [ты любишь ходить в школу ?]
    [ты хочешь учиться на "5" ?]
    [какого цвета снег ?]
    [какого цвета слово "нет" ?]
    [ты образованный человек ?]
    [ в каком платье / костюме ты будешь на свадьбе ? ]
]
путь "реплика [
    [молодец !]
    [а ты , ещё умеешь писать ?]
    [ну надо же !]
    [да ты что ! ]
    [ты , ЮЛИЙ ЦЕЗАРЬ !]
]
всегда [
спроси элемент :х :вопрос
путь "х :х + 1
если входит? ответ :зс [сообщи[извини, но ты не справился , пока !]автостоп]
сообщи элемент (1 + сл 5) :реплика
если :х > 6 [сообщи [молодец ! ты справился с заданием , пока !] автостоп]
]
конец
```



Наши авторы, 2004.
Our authors, 2004.

*Волкова РизидА Анверовна,
руководитель группы информационных
технологий общеобразовательного
учебного заведения Тольяттинской
академии управления.*