

Программирование (АлгЯзык)

§ 19. Символьные строки

§ 20. Обработка массивов

§ 21. Матрицы (двумерные массивы)

§ 22. Сложность алгоритмов

§ 23. Как разрабатывают программы? § 23.
Как разрабатывают программы?

§ 24. Процедуры

§ 25. Функции

Программирование (АлгЯзык)

§ 19. Символьные строки

Что такое символьная строка?

Символьная строка – это последовательность СИМВОЛОВ.

Хочется:

- строка – единый объект
- длина строки может меняться во время работы программы

лит s | **символьная строка**

литерный тип

Символьные строки

Присваивание:

```
s := 'Вася пошёл гулять'
```

ЛИТ S

Ввод с клавиатуры:

```
ВВОД S
```

Вывод на экран:

```
ВЫВОД S
```

Длина строки:

```
цел n
```

```
n := ДЛИН (s)
```

Сравнение строк

лит s

вывод 'Введите пароль: '

ввод s

если s = 'sEzAm' **то**

вывод 'Слушаюсь и повинуюсь!'

иначе

вывод 'Пароль неправильный'

все



Какой правильный пароль?



Как одна строка может быть меньше другой?

стоит раньше в отсортированном списке

Сравнение строк

```

лит s
s1 := 'паровоз'
s2 := 'пароход'
если s1 < s2 то
    вывод s1, ' < ', s2
иначе
    если s1 = s2 то
        вывод s1, ' = ', s2
    иначе
        вывод s1, ' > ', s2
все
все

```



Что выведет?

паровоз < пароход



В < Х!

«В»: КОД 226

«Х»: КОД 245

первые отличающиеся
буквы

Сравниваем с начала: **паровоз**
пароход

Посимвольная обработка строк

```
s[4] := 'a'
```

Задача. Ввести строку и заменить в ней все буквы «э» на буквы «е».

```
цел i  
нц для i от 1 до длин(s)  
    если s[i]='э' то  
        s[i]:='е'  
    все  
кц
```

для каждого символа
строки

Задачи

«А»: Напишите программу, которая вводит строку, состоящую только из точек и букв X, и заменяет в ней все точки на нули и все буквы X на единицы.

Пример:

Введите строку: **..X.XX.**

Двоичный код: 0010110

«В»: Напишите программу, которая в символьной строке заменяет все нули на единицы и наоборот. Остальные символы не должны измениться.

Пример:

Введите строку: **10a01Vx1010c**

Инверсия: 01a10Vx0101c

Задачи

«С»: Введите битовую строку и дополните её последним битом, который должен быть равен 0, если в исходной строке чётное число единиц, и равен 1, если нечётное (в получившейся строке должно всегда быть чётное число единиц).

Пример:

Введите битовую строку: **01101010110**

Результат: 011010101100

Операции со строками

Объединение (конкатенация) :

```
s1 := 'Привет'
```

```
s2 := 'Вася'
```

```
s := s1 + ', ' + s2 + '!'
```

'Привет, Вася!'

Срез:

```
s := '123456789'
```

```
s1 := s[3:7] | '34567'
```

с какого
СИМВОЛА

до какого
СИМВОЛА

Операции со строками

Удаление:

```
s := '123456789'
```

```
удалить (s, 3, 6) | '129'
```

с какого
СИМВОЛА

СКОЛЬКО
СИМВОЛОВ

Вставка:

```
s := '123456789'
```

```
вставить ('ABC', s, 3) | '12ABC3456789'
```

что

куда

с какого
СИМВОЛА



Процедуры или функции?

Поиск в строках

```
s := 'Здесь был Вася.'
```

что где

```
n := позиция('с', s)
```

```
если n > 0 то
```

```
    вывод 'Номер символа ', n
```

```
иначе
```

```
    вывод 'Символ не найден.'
```

```
все
```



Находит первое слева вхождение подстроки!

Задачи

«А»: Ввести с клавиатуры в одну строку фамилию и имя, разделив их пробелом. Вывести первую букву имени с точкой и потом фамилию.

Пример:

Введите фамилию и имя:

Иванов Петр

П. Иванов

«В»: Ввести с клавиатуры в одну строку фамилию, имя и отчество, разделив их пробелом. Вывести фамилию и инициалы.

Пример:

Введите фамилию, имя и отчество:

Иванов Петр Семёнович

П.С. Иванов

Задачи

«С»: Ввести адрес файла и «разобрать» его на части, разделенные знаком ' / '. Каждую часть вывести в отдельной строке.

Пример:

Введите адрес файла:

C: /фото/2015/Байкал/shaman.jpg

C:

фото

2015

Байкал

shaman.jpg

Преобразования «строка» → «число»

Целое число:

да или нет

цел N, **лит** s, **лог** ОК

s := '123'

N := **лит_в_цел**(s, ОК) | N = 123

если **не** ОК то вывод 'Ошибка!' все

Вещественное число:

вещ X, **лит** s, **лог** ОК

s := '123.456';

X := **лит_в_вещ**(s, ОК) | X = 123.456

если **не** ОК то вывод 'Ошибка!' все

Преобразования «число» → «строка»

цел N, **вещ** X, **лит** s

N := 123

s := **цел_в_лит**(N) | '123'

X := 123.456

s := **вещ_в_лит**(X) | '123.456'



Как объявить переменные?

Задачи

«А»: Напишите программу, которая вычисляет сумму двух чисел, введенную в форме символьной строки. Все числа целые.

Пример:

Введите выражение :

12+3

Ответ: 15

«В»: Напишите программу, которая вычисляет сумму трёх чисел, введенную в форме символьной строки. Все числа целые.

Пример:

Введите выражение :

12+3+45

Ответ: 60

Задачи

«С»: Напишите программу, которая вычисляет сумму произвольного количества чисел, введенную в форме символьной строки. Все числа целые.

Пример:

Введите выражение :

12+3+45+10

Ответ: 70

«D»: Напишите программу, которая вычисляет выражение, содержащее целые числа и знаки сложения и вычитания.

Пример:

Введите выражение :

12+134-45-17

Ответ: 84

Программирование (АлгЯзык)

§ 20. Обработка массивов

Обработка потока данных

Задача. С клавиатуры вводятся числа, ввод завершается числом 0. Определить, сколько было введено положительных чисел.

- 1) нужен счётчик
- 2) счётчик увеличивается
- 3) нужен цикл
- 4) это цикл с условием (число шагов неизвестно)

 ?

Когда увеличивать счётчик?

 ?

Какой цикл?

счётчик = 0

пока не введён 0 :

если введено число > 0 то

счётчик := счётчик + 1

Обработка потока данных

```
цел x, count
```

```
count := 0
```

```
ввод x
```

```
нц пока x <> 0
```

откуда взять x?

```
    если x > 0 то
```

```
        count := count + 1
```

```
    все
```

```
    ввод x
```



Что плохо?

```
кц
```

```
вывод count
```

Найди ошибку!

```
цел x, count
count := 0
ввод x
нц пока x <> 0
    если x > 0 то
        count := count + 1
    все
клввод x
вывод count
```

Найди ошибку!

```
цел x, count
```

```
count := 0
```

```
нц пока x = 0
```

```
  если x ><> то
```

```
    count := count + 1
```

```
  все
```

```
ввод x
```

```
кц
```

```
вывод count
```

Обработка потока данных

Задача. С клавиатуры вводятся числа, ввод завершается числом 0. Найти сумму введённых чисел, оканчивающихся на цифру "5".

- 1) нужна переменная для суммы
- 2) число добавляется к сумме, если оно заканчивается на "5"
- 3) нужен цикл с условием

сумма := 0

пока не введён 0:

если число оканчивается на "5" то

сумма := сумма + число

если $\text{mod}(x, 10) = 5$ то



Как это записать?

Обработка потока данных

Задача. С клавиатуры вводятся числа, ввод завершается числом 0. Найти сумму введённых чисел, оканчивающихся на цифру "5".

```
цел x, sum
```

```
sum := 0
```

```
ввод x
```

```
нц пока x <> 0
```

```
    если mod(x, 10) = 5 то
```

```
        sum := sum + x
```

```
    все
```

```
    ввод x
```

```
кц
```

```
вывод sum
```



Чего не хватает?

Найди ошибку!

```
цел x, sum
```

```
sum := 0
```

```
ввод x а x <> 0
```

```
если mod(x, 10) = 5 то
```

```
    sum := sum + x
```

```
все
```

```
ввод x
```

```
кц
```

```
вывод sum
```

Задачи

- «А»: На вход программы поступает неизвестное количество целых чисел, ввод заканчивается нулём. Определить, сколько получено чисел, которые делятся на 3.
- «В»: На вход программы поступает неизвестное количество целых чисел, ввод заканчивается нулём. Определить, сколько получено двузначных чисел, которые заканчиваются на 3.

Задачи

- «С»: На вход программы поступает неизвестное количество целых чисел, ввод заканчивается нулём. Найти среднее арифметическое всех двузначных чисел, которые делятся на 7.
- «D»: На вход программы поступает неизвестное количество целых чисел, ввод заканчивается нулём. Найти максимальное из введённых чётных чисел.

Перестановка элементов массива



Как поменять местами значения двух переменных a и b ?

вспомогательная
переменная

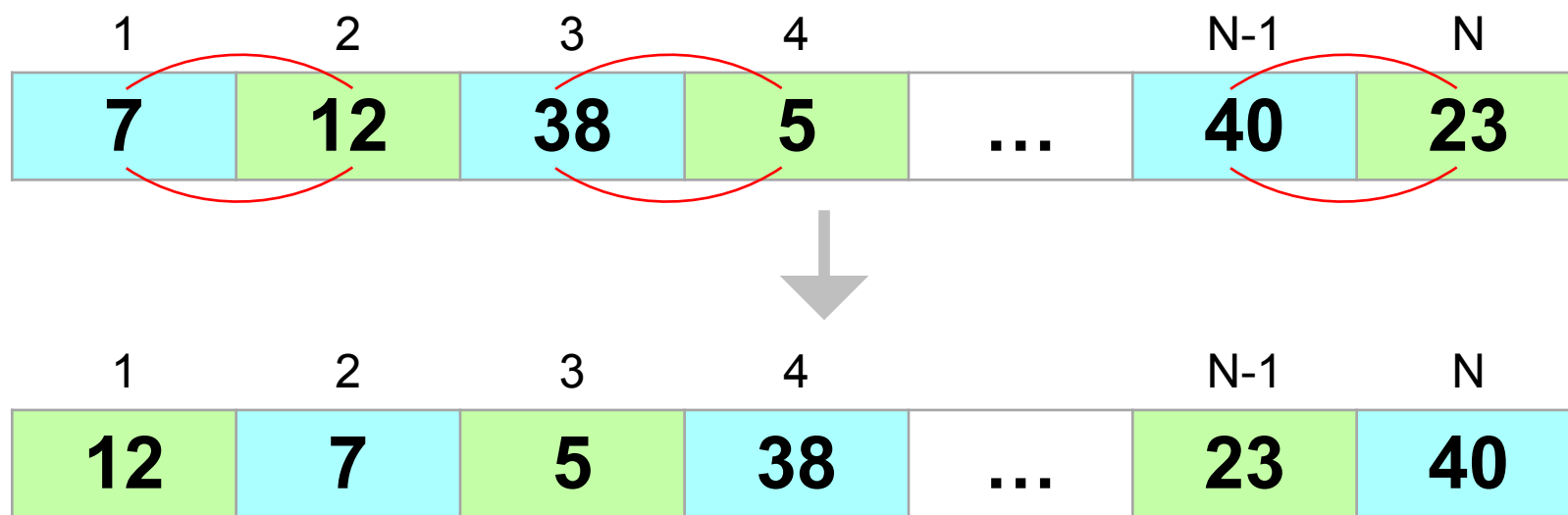
```
c := a
a := b
b := c
```

элементы массива:

```
c := A[i]
A[i] := A[k]
A[k] := c
```

Перестановка пар соседних элементов

Задача. Массив A содержит чётное количество элементов N . Нужно поменять местами пары соседних элементов: первый со вторым, третий — с четвёртым и т. д.



Перестановка пар соседних элементов

нц для i от 1 до N

поменять местами $A[i]$ и $A[i+1]$

кц



Что плохо?

1	2	3	4	5	6
7	12	38	5	40	23
12	7	38	5	40	23
12	38	7	5	40	
12	38	5	7	40	23
12	38	5	40	7	23
12	38	5	40	23	7

выход за границы массива



Перестановка пар соседних элементов

не выходим за
границу

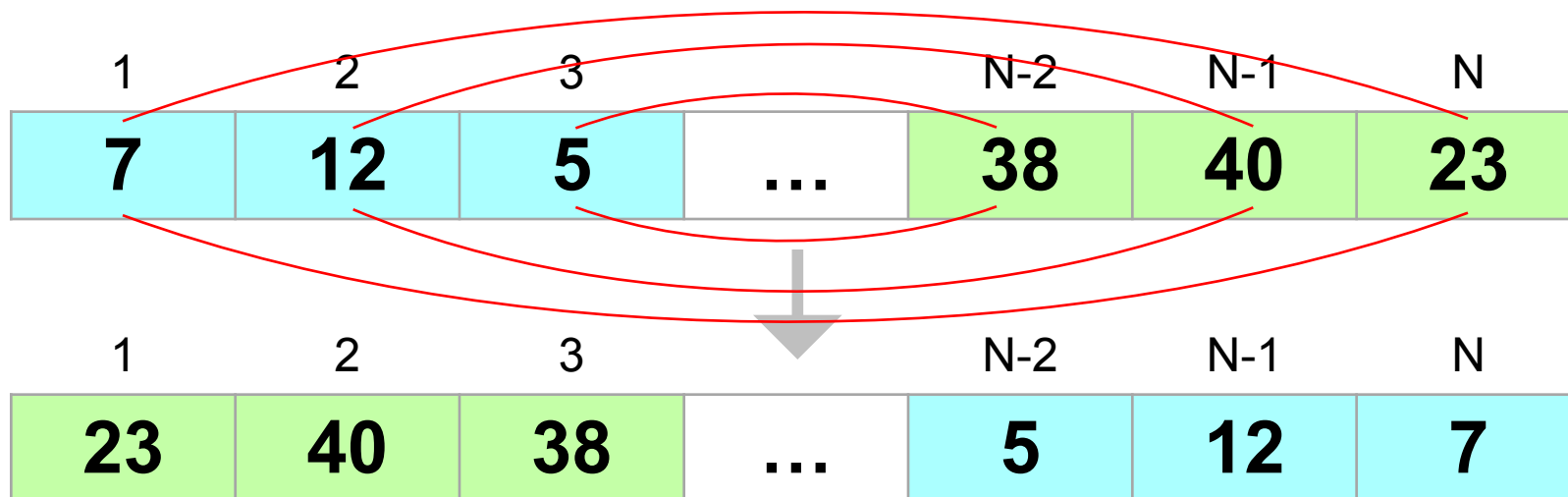
не трогаем те, что
уже переставлены

```
нц для i от 1 до N-1 шаг 2
  | переставляем A[i] и A[i+1]
  c := A[i]
  A[i] := A[i+1]
  A[i+1] := c
кц
```

$A[1] \leftrightarrow A[2], A[3] \leftrightarrow A[4], \dots, A[N-1] \leftrightarrow A[N]$

Реверс массива

Задача. Переставить элементы массива в обратном порядке (выполнить *реверс*).



$$A[1] \leftrightarrow A[N]$$

$$1+N = N+1$$

$$A[2] \leftrightarrow A[N-1]$$

$$2+N-1 = N+1$$

$$A[i] \leftrightarrow A[N+1-i]$$

$$i+??? = N+1$$

$$A[N] \leftrightarrow A[1]$$

$$N+1 = N+1$$

Реверс массива

нц для i от 1 до $\text{div}(N, 2)$

поменять местами $A[i]$ и $A[N+1-i]$

кц



Что плохо?

1	2	3	4
7	12	40	23
23	12	40	7
23	40	12	7
23	12	40	7
7	12	40	23

$i=1$

$i=2$

$i=3$

$i=4$



Как исправить?

Линейный поиск в массиве

Задача. Найти в массиве элемент, равный X , и его номер.

$X = 5$

	1	2	3	4	5	6
	7	12	38	5	40	23

```
i := 1
```

```
нц пока A[i] <> X
```

```
  i := i + 1
```

```
кц
```

```
вывод 'A[' , i , ']' = ' , X
```



Что плохо?



Если искать 4?



Нельзя выходить за границы массива!

Линейный поиск в массиве

```
i := 1
нц пока i <= N и A[i] <> X
    i := i + 1
кц
если i <= N то
    вывод 'A[', i, ']=', X
иначе
    вывод 'Не нашли!'
все
```



Как проверить, нашли или нет?

Досрочный выход из цикла

Задача. Найти в массиве элемент, равный X , и его номер.

$nX := 0$ | номер элемента

цел nX

нц для i от 1 до N

если $A[i] = X$ то нашли!

$nX := i$ | запомнить номер

выход

все

сразу выйти
из цикла

кц

если $nX > 0$ то

вывод 'A[', nX , ']=', X

иначе

вывод 'Не нашли!'

все



Как объявить nX ?

Задачи

«А»: Напишите программу, которая заполняет массив из $N = 10$ элементов случайными числами в диапазоне $[0,20]$, выводит его на экран, а затем находит индекс первого элемента, равного введённому числу X . Программа должна вывести ответ «не найден», если в массиве таких элементов нет.

Пример:

Массив: 5 16 2 13 3 14 18 13 16 9

Что ищем: 13

$A[4] = 13$

Задачи

«В»: Напишите программу, которая заполняет массив из $N = 10$ элементов случайными числами в диапазоне $[-10, 10]$, выводит его на экран, а затем находит индекс **последнего** элемента, равного введённому числу X . Программа должна вывести ответ «не найден», если в массиве таких элементов нет.

Пример:

Массив: -5 -6 2 3 -3 0 8 -3 0 9

Что ищем: 0

$A[9] = 0$

Задачи

«С»: Напишите программу, которая заполняет массив из $N = 10$ элементов случайными числами в диапазоне $[10, 50]$, выводит его на экран, а затем находит индексы всех элементов, равных введённому числу X . Программа должна вывести ответ «не найден», если в массиве таких элементов нет.

Пример:

Массив: 12 45 30 18 30 15 30 44 32 17

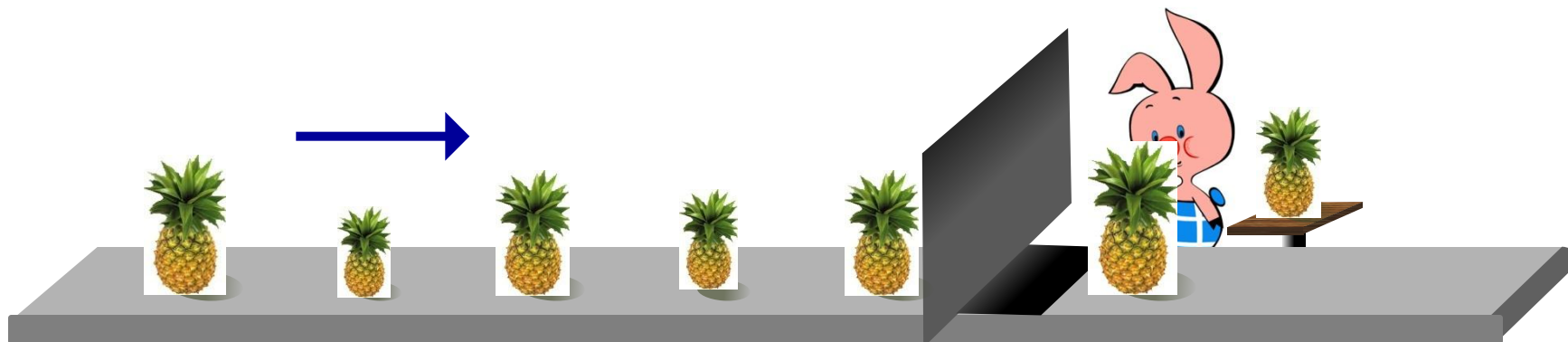
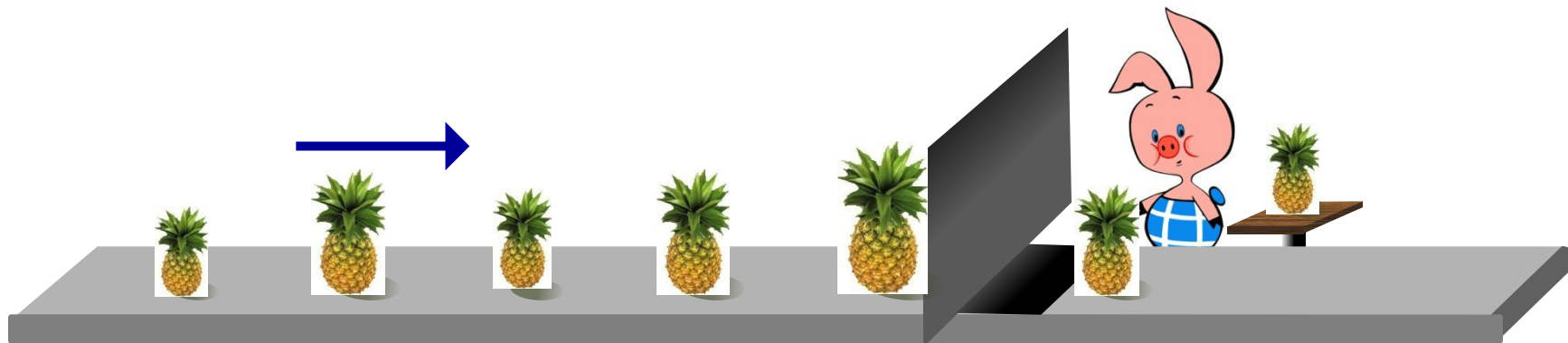
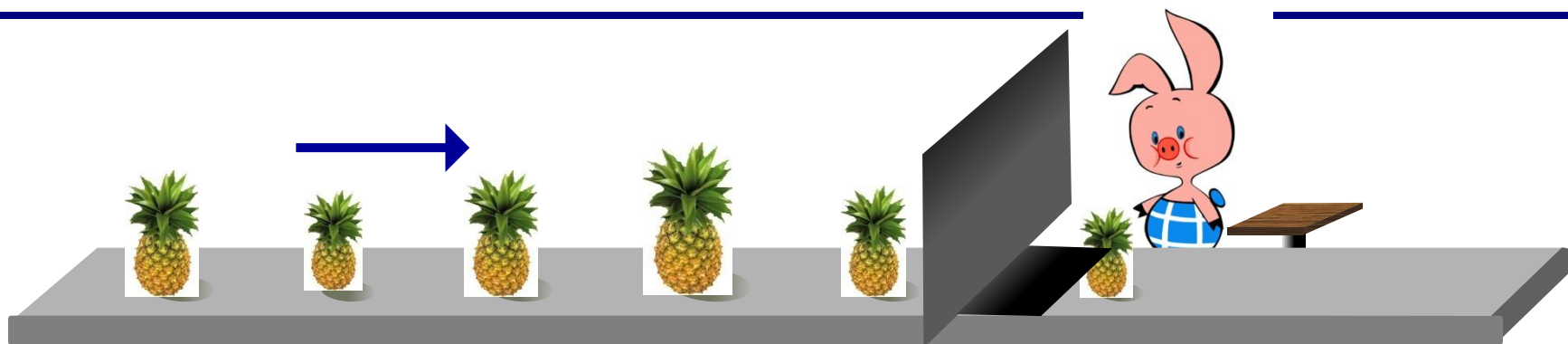
Что ищем: 30

$A[3] = 30$

$A[5] = 30$

$A[7] = 30$

Поиск максимального элемента



Поиск максимального элемента

? Какие переменные нужны?

```
нц для i от 1 до N
  если A[i] > M то
    M := A[i]
все
кц
вывод M
```

? Чего не хватает?

? Какое начальное значение взять для M?

1) M – значение, которое заведомо меньше всех элементов массива

или

2) $M = A[1]$ (или любой другой элемент)

максимальный не меньше, чем $A[1]$

Поиск максимального элемента

```
М := А[1]
нц для і от 2 до N
  если А[і] > М то
    М := А[і]
  все
кц
вывод М
```

начинаем с А[2], так как А[1] мы уже посмотрели



Как найти минимальный?

Номер максимального элемента

Задача. Найти в массиве максимальный элемент и его номер.

? Какие переменные нужны?

```
М := A[1]; nMax := 1
нц для i от 2 до N
  если A[i] > М то
    М := A[i]
    nMax := i
все
кц
вывод 'A[', nMax, ']=', М
```

? Можно ли убрать одну переменную?

Номер максимального элемента

! Если знаем $nMax$, то $M=A[nMax]$!

```
M := A[1]; nMax := 1  
нц для i от 2 до N  
  если  $A[i] > A[nMax]$  то  
    M := A[i]  
    nMax := i  
все  
кц  
вывод 'A[', nMax, '= ',  $A[nMax]$ 
```

Максимальный не из всех

Задача. Найти в массиве максимальный из отрицательных элементов.

```
М := А[1]
нц для i от 2 до N
  если А[i] < 0 и А[i] > М то
    М := А[i]
  все
кц
вывод М
```



Что плохо?



Как исправить?

1	2	3	4	5
5	-2	8	3	-1

М = 5

Максимальный не из всех

Задача. Найти в массиве максимальный из отрицательных элементов.

```
М := А[1]
нц для i от 2 до N
  если А[i] < 0 то
    если М >= 0 или А[i] > М то
      М := А[i]
    все
  все
кц
вывод М
```

сначала записали неотрицательный!



Если нет отрицательных?

Задачи

- «А»: Напишите программу, которая заполняет массив из 20 элементов случайными числами на отрезке [50; 150] и находит в нём минимальный и максимальный элементы и их номера.
- «В»: Напишите программу, которая получает с клавиатуры значения элементов массива и выводит количество элементов, имеющих максимальное значение.
- «С»: Напишите программу, которая заполняет массив из 20 элементов случайными числами на отрезке [100; 200] и находит в нём пару соседних элементов, сумма которых минимальна.

Задачи

«D»: Напишите программу, которая заполняет массив из 20 элементов случайными числами на отрезке $[-100; 100]$ и находит в каждой половине массива пару соседних элементов, сумма которых максимальна.

Задачи-2 (максимум в потоке)

- «А»: На вход программы поступает неизвестное количество целых чисел, ввод заканчивается нулём. Напишите программу, которая находит минимальное и максимальное среди полученных чисел.
- «В»: На вход программы поступает неизвестное количество целых чисел, ввод заканчивается нулём. Напишите программу, которая находит минимальное число, делящееся на 3, среди полученных чисел.
- «С»: На вход программы поступает неизвестное количество чисел целых, ввод заканчивается нулём. Напишите программу, которая находит максимальное двузначное число, заканчивающееся на 6, среди полученных чисел.

Задачи-2 (максимум в потоке)

«D»: На вход программы поступает неизвестное количество чисел целых, ввод заканчивается нулём. Напишите программу, которая находит среди полученных чисел пару полученных друг за другом чисел, сумма которых максимальна.

Сортировка

Сортировка — это расстановка элементов списка (массива) в заданном порядке.

Задача. Отсортировать элементы в порядке **возрастания** (*неубывания* – если есть одинаковые).

Алгоритмы сортировки:

- простые, но медленные (при больших N)
- быстрые, но сложные...

Сортировка выбором

? Где должен стоять минимальный элемент?

- нашли минимальный, поставили его на первое место

```
c := A[nMin]
```

```
A[nMin] := A[1]
```

```
A[1] := c
```

? Как?

? Что дальше?

- из оставшихся нашли минимальный, поставили его на второе место и т.д.

5	-2	8	3	-1
-2	5	8	3	-1
-2	-1	8	3	5

Сортировка выбором

```
нц для  $i$  от 1 до  $N-1$   
  | ищем минимальный среди  $A[i]..A[N]$ 
```

не трогаем те, которые
уже поставлены

```
  nMin :=  $i$ 
```

```
  нц для  $j$  от  $i+1$  до  $N$ 
```

```
    если  $A[j] < A[nMin]$  то
```

```
      nMin :=  $j$ 
```

```
    все
```

```
  кц
```

```
  | переставляем  $A[i]$  и  $A[nMin]$ 
```

```
   $c := A[i]$ 
```

```
   $A[i] := A[nMin]$ 
```

```
   $A[nMin] := c$ 
```

```
кц
```



Почему цикл до $N-1$?

Задачи

«А»: Напишите программу, которая заполняет массив из $N = 10$ элементов случайными числами в диапазоне $[0,20]$ и сортирует его в порядке убывания.

Пример:

Массив: 5 16 2 13 3 14 18 13 16 9

Сортировка: 18 16 16 14 13 13 9 5 3 2

«В»: Напишите программу, которая заполняет массив из $N = 10$ элементов случайными числами в диапазоне $[10,100]$ и сортирует его по возрастанию последней цифры числа (сначала идут все числа, которые заканчиваются на 0, потом все, которые заканчиваются на 1, и т.д.).

Пример:

Массив: 12 10 31 40 55 63 28 87 52 92

Сортировка: 10 40 31 12 52 92 63 55 87 28

Задачи

«С»: Напишите программу, которая заполняет массив из $N = 10$ элементов случайными числами в диапазоне $[0,20]$ и сортирует его в порядке возрастания. На каждом шаге цикла выполняется поиск максимального (а не минимального!) элемента.

Пример:

Массив: 5 16 2 13 3 14 18 13 16 9

Сортировка: 2 3 5 9 13 13 14 16 16 18

Программирование (АлгЯзык)

§ 21. Матрицы (двумерные массивы)

Что такое матрица?

	○	×
	○	×
○	×	

нет знака

НОЛИК

крестик

строка 2,
столбец 3



Как закодировать?

Матрица — это прямоугольная таблица, составленная из элементов одного типа (чисел, строк и т.д.).

Каждый элемент матрицы имеет два индекса — номера строки и столбца.

Объявление матриц

цел $N = 3$, $M = 4$

целтаб $A[1:N, 1:M]$

вещтаб $X[-3:0, -8:M]$

ло: строки $1:N$, столбцы

строки

столбцы

Простые алгоритмы

Заполнение случайными числами:

```
нц для i от 1 до N
  нц для j от 1 до M
    A[i,j] := irand(20,80)
    вывод A[i,j], ' '
  кц
вывод нс
кц
```



Вложенный цикл!

в одной строке
через пробел

следующий – с
НОВОЙ строки

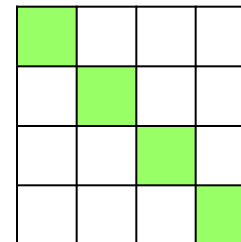
Суммирование:

```
sum := 0
нц для i от 1 до N
  нц для j от 1 до M
    sum := sum + A[i,j]
  кц
кц
```

Перебор элементов матрицы

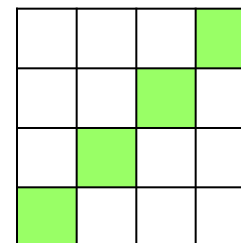
Главная диагональ:

```
нц для i от 1 до N
  | работаем с A[i,i]
кц
```



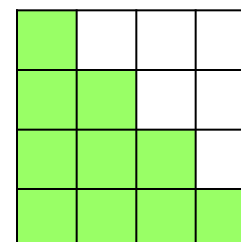
Побочная диагональ:

```
нц для i от 1 до N
  | работаем с A[i,N+1-i]
кц
```



Главная диагональ и под ней:

```
нц для i от 1 до N
  нц для j от 1 до i
    | работаем с A[i,j]
  кц
кц
```



Перестановка строк

2-я и 4-я строки:

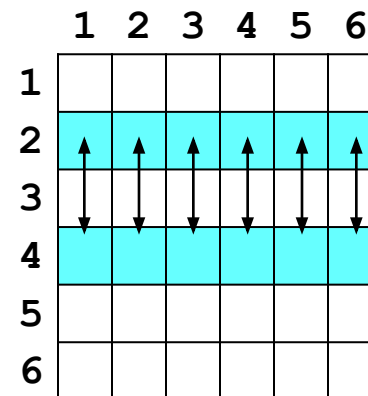
нц для j от 1 до M

$c := A[2, j]$

$A[2, j] := A[4, j]$

$A[4, j] := c$

кц



Задачи

«А»: Напишите программу, которая заполняет матрицу случайными числами и находит максимальный элемент на главной диагонали квадратной матрицы.

Пример:

Матрица А:

12 34 14 65

71 88 23 45

87 46 53 39

76 58 24 92

Результат: $A[4,4] = 92$

Задачи

«В»: Напишите программу, которая заполняет матрицу случайными числами и находит максимальный элемент матрицы и его индексы (номера строки и столбца).

Пример:

Матрица А:

12 34 14 65

71 88 23 98

87 46 53 39

76 58 24 92

Максимум: $A[2, 4] = 98$

Задачи

«С»: Напишите программу, которая заполняет матрицу случайными числами и находит минимальный из чётных положительных элементов матрицы. Учтите, что таких элементов в матрице может и не быть.

Пример:

Матрица A:

16 34 14 65

71 88 23 45

87 12 53 39

76 58 24 92

Результат: $A[3,2] = 12$

Программирование (АлгЯзык)

§ 22. Сложность алгоритмов

Как сравнивать алгоритмы?

- быстродействие (**временная сложность**)
- объём требуемой памяти (**пространственная сложность**)
- понятность



Обычно не бывает все хорошо!

Время работы алгоритма – это количество элементарных операций T , выполненных исполнителем.

Функция $T(N)$ называется

временной сложностью алгоритма

зависит от
количества данных
(размера массива N)

$$T(N) = 2N^3$$



Как увеличится время работы при увеличении N в 10 раз?

Примеры определения сложности

Задача 1. Вычислить сумму первых трёх элементов массива (при $N \geq 3$).

```
Sum := A[1] + A[2] + A[3]
```

$$T(N) = 3$$

2 сложения
+ запись в
память

Задача 2. Вычислить сумму всех элементов массива.

```
Sum := 0
```

```
нц для i от 1 до N
```

```
    Sum := Sum + A[i]
```

```
кц
```

$$T(N) = 2N + 1$$

N сложений, $N+1$
операций записи

Примеры определения сложности

Задача 3. Отсортировать все элементы массива по возрастанию методом выбора.

```
нц для i от 1 до N-1
  nMin := i
  нц для j от i+1 до N
    если A[i] < A[nMin] то nMin := j все
  кц
  c := A[i]; A[i] := A[nMin]; A[nMin] := c
кц
```

Число сравнений:

$$T_c(N) = (N-1) + (N-2) + \dots + 2 + 1 = \frac{N(N-1)}{2} = \frac{1}{2}N^2 - \frac{1}{2}N$$

Число перестановок: $T_n(N) = N - 1$

Примеры определения сложности

Задача 4. Найти сумму элементов квадратной матрицы размером $N \times N$.

```
Sum := 0
нц для i от 1 до N
  нц для j от 1 до N
    Sum := Sum + A[i, j]
  кц
кц
```



Самостоятельно! 😊

Сравнение алгоритмов по сложности

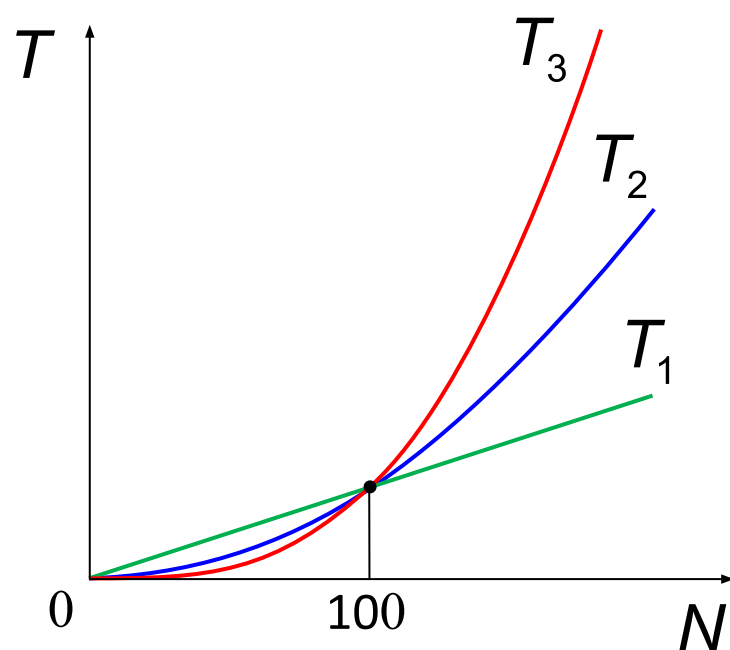
$$T_1(N) = 10000 \cdot N$$

$$T_2(N) = 100 \cdot N^2$$

$$T_3(N) = N^3$$



Какой алгоритм выбрать?



при $N < 100$:

$$T_3(N) < T_2(N) < T_1(N)$$

при $N > 100$:

$$T_3(N) > T_2(N) > T_1(N)$$



Нужно знать размер данных!

Асимптотическая сложность

Асимптотическая сложность – это оценка скорости роста количества операций при больших значениях N .

линейная

постоянная

сложность $O(N)$ $\Leftrightarrow T(N) \leq c \cdot N$ для $N \geq N_0$

сумма элементов массива:

$$T(N) = 2 \cdot N - 1 \leq 2 \cdot N \text{ для } N \geq 1 \Rightarrow O(N)$$

квадратичная

сложность $O(N^2)$ $\Leftrightarrow T(N) \leq c \cdot N^2$ для $N \geq N_0$

сортировка методом выбора:

$$T_c(N) = \frac{1}{2} N^2 - \frac{1}{2} N \leq \frac{1}{2} N^2 \text{ для } N \geq 0 \Rightarrow O(N^2)$$

Асимптотическая сложность

кубическая

сложность $O(N^3)$ $\Leftrightarrow T(N) \leq c \cdot N^3$ для $N \geq N_0$

сложность $O(2^N)$

сложность $O(N!)$

задачи оптимизации,
полный перебор вариантов

Факториал числа N : $N! = 1 \cdot 2 \cdot 3 \dots$

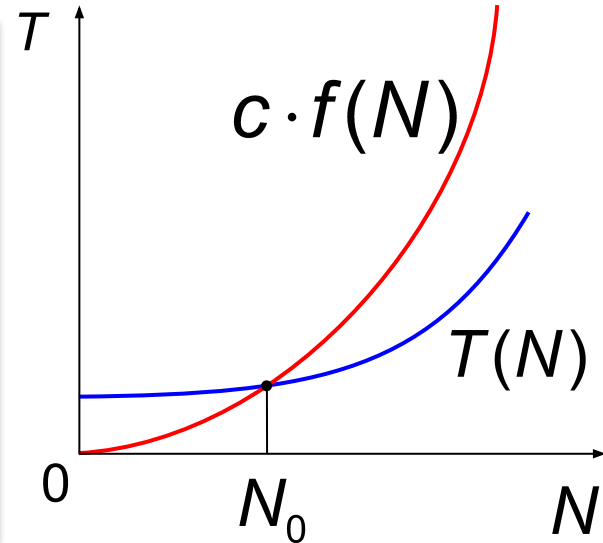
N	$T(N)$	время выполнения
	N	100 нс
	N^2	10 мс
	N^3	0,001 с
	2^N	10^{13} лет

$N = 100,$
1 млрд оп/с

Асимптотическая сложность

Алгоритм относится к классу $O(f(N))$, если найдется такая постоянная c , что начиная с некоторого $N = N_0$ выполняется условие

$$T(N) \leq c \cdot f(N)$$



это верхняя оценка!

$$O(N) \Rightarrow O(N^2) \Rightarrow O(N^3) \Rightarrow O(2^N)$$

«Алгоритм имеет сложность $O(N^2)$ ».

обычно — наиболее точная верхняя оценка!

Программирование (АлгЯзык)

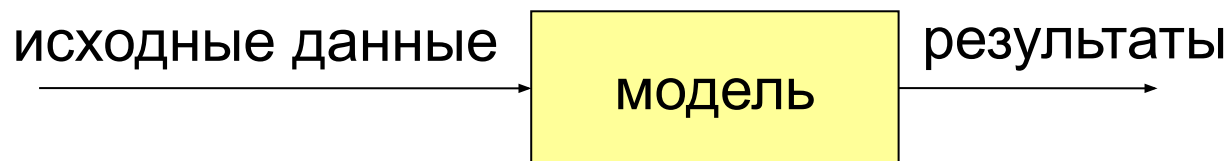
§ 23. Как разрабатывают программы

Этапы разработки программ

I. Постановка задачи

Документ: *техническое задание*.

II. Построение модели



Формализация: запись модели в виде формул (на формальном языке).

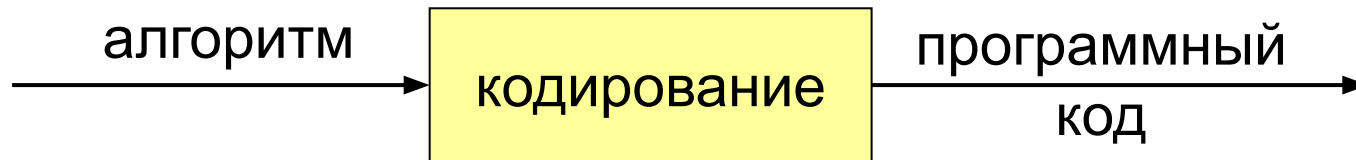
III. Разработка алгоритма и способа хранения данных

«Алгоритмы + структуры данных = программы»
(Н. Вирт)

Этапы разработки программ

IV. Кодирование

Запись алгоритма на языке программирования.



V. Отладка

Поиск и исправление ошибок в программах:

- **синтаксические** – нарушение правил языка программирования
 - **логические** – ошибки в алгоритме
- могут приводить к **отказам** – аварийным ситуациям во время выполнения (*run-time error*)

Этапы разработки программ

VI. Тестирование

Тщательная проверка программы во всех режимах:

- **альфа-тестирование** – внутри компании (тестировщики)
- **бета-тестирование** – (доверенные) пользователи

VII. Документирование

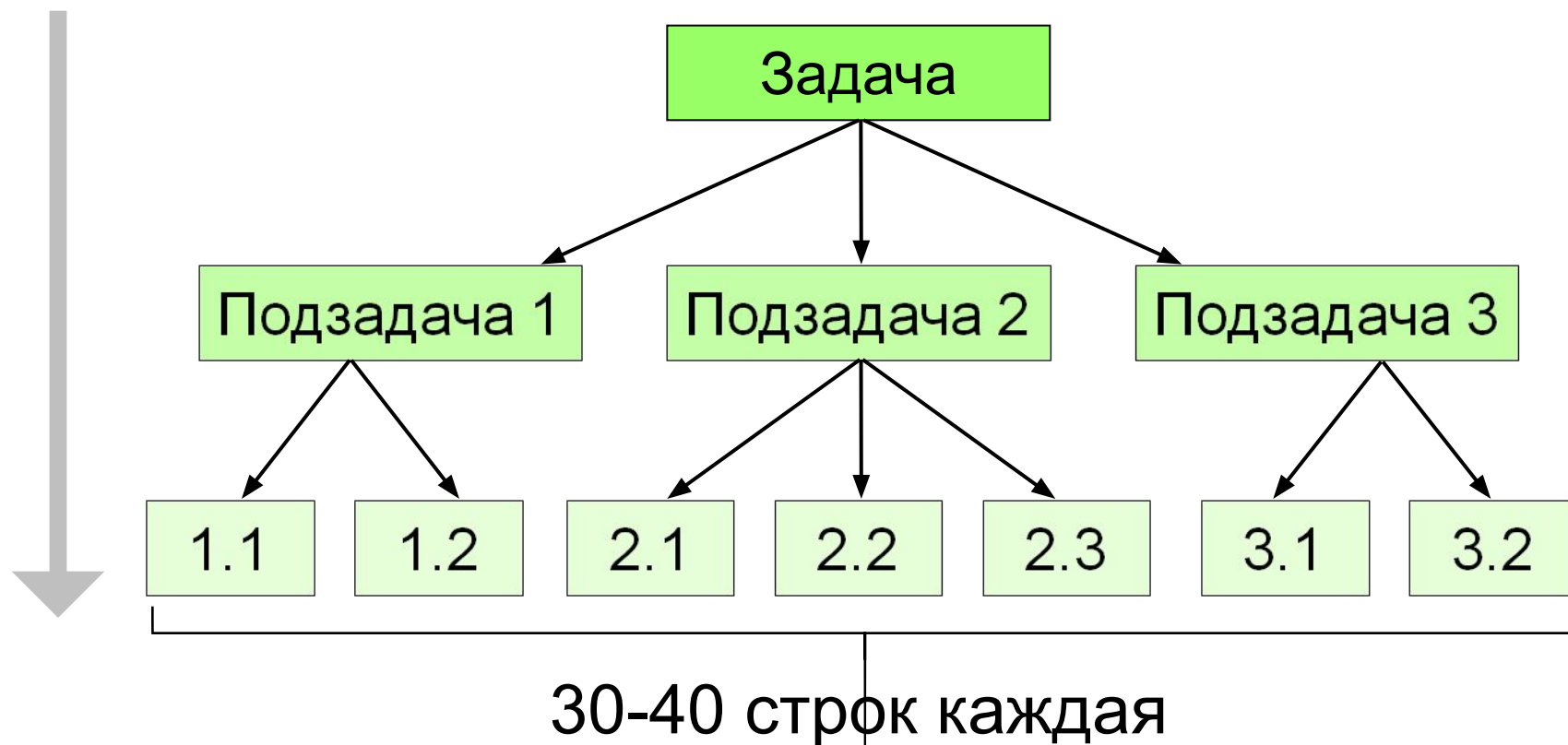
Технические писатели

VIII. Внедрение и сопровождение

- обучение пользователей
- исправление найденных ошибок
- техподдержка

Методы проектирования программ

«Сверху вниз» (последовательное уточнение)



Методы проектирования программ

«Сверху вниз» (последовательное уточнение)



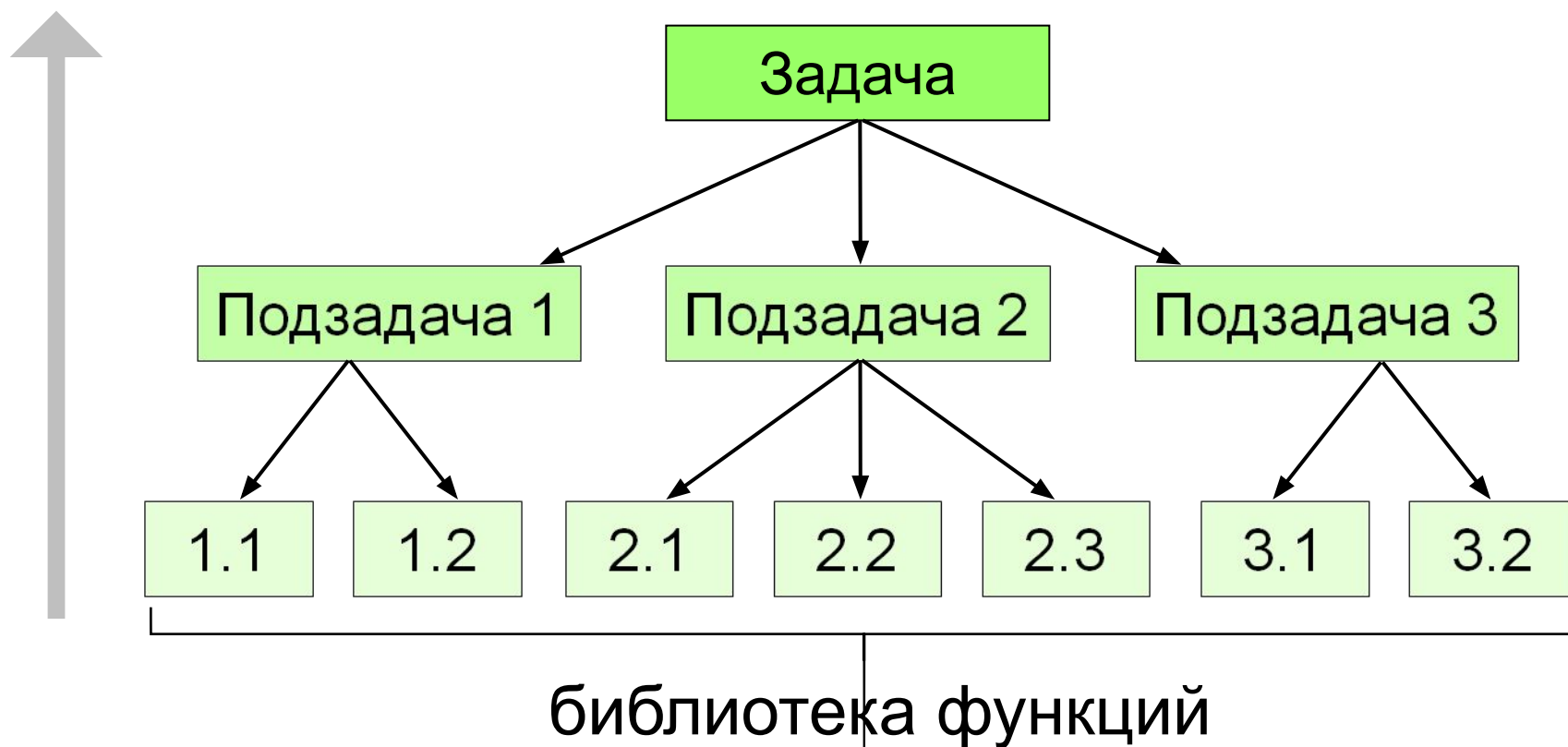
- сначала задача решается «в целом»
- легко распределить работу
- легче отлаживать программу (всегда есть полный работающий вариант)



- в нескольких подзадачах может потребоваться решение одинаковых подзадач нижнего уровня
- быстроедействие не известно до последнего этапа (определяется нижним уровнем)

Методы проектирования программ

«Снизу вверх» (восходящее)



Методы проектирования программ

«Снизу вверх» (восходящее)



- нет дублирования
- сразу видно быстрое действие



- сложно распределять работу
- сложнее отлаживать (увеличение числа связей)
- плохо видна задача «в целом», может быть нестыковка на последнем этапе



Почти всегда используют оба подхода!

Отладка программы

Программа решения квадратного уравнения

$$ax^2 + bx + c = 0$$

```
алг КвУр
```

```
нач
```

```
  вещь a, b, c, D, x1, x2
```

```
  вывод 'Введите a, b, c: '
```

```
  ввод a, b, c
```

```
  D:=b*b-4*a*a
```

```
  x1:=(-b+sqrt(D))/2*a
```

```
  x2:=(-b-sqrt(D))/2*a
```

```
  вывод 'x1=', x1, ' x2=', x2
```

```
кон
```

Тестирование

Тест 1. $a = 1, b = 2, c = 1.$

Ожидание:

$x1=-1.0 \quad x2=-1.0$

Реальность:

$x1=-1.0 \quad x2=-1.0$



Тест 2. $a = 1, b = -5, c = 6.$

$x1=3.0 \quad x2=2.0$

$x1=4.791 \quad x2=0.209$



Найден вариант, когда программа работает неверно.
Ошибка **воспроизводится!**

Возможные причины:

- неверный ввод данных
- неверное вычисление дискриминанта
- неверное вычисление корней
- неверный вывод результатов

$$D = b^2 - 4ac$$

$$x_{1,2} = \frac{-b \pm \sqrt{D}}{2a}$$

Отладочная печать

Идея: выводить все промежуточные результаты.

```
ввод a, b, c
```

```
вывод a, ' ', b, ' ', c, нс
```

```
D := b*b - 4*a*a
```

```
вывод 'D=' , D, нс
```

```
...
```

Результат:

```
Введите a, b, c: 1 -5 6
```

```
1.0 -5.0 6.0
```

```
D=21.0
```

$$D = b^2 - 4ac = 25 - 4 \cdot 1 \cdot 6 = 1$$

```
D := b*b - 4*a* c ;
```



Одна ошибка найдена!

Отладка программы

Тест 1. $a = 1, b = 2, c = 1.$

Ожидание:

$x1 = -1.0 \quad x2 = -1.0$

Реальность:

$x1 = -1.0 \quad x2 = -1.0$



Тест 2. $a = 1, b = -5, c = 6.$

$x1 = 3.0 \quad x2 = 2.0$

$x1 = 3.0 \quad x2 = 2.0$



Программа работает верно?

Тест 3. $a = 8, b = -6, c = 1.$

$x1 = 0.5 \quad x2 = 0.25$

$x1 = 32.0 \quad x2 = 16.0$



$x1 := (-b + \text{sqrt}(D)) / (2 * a)$

$x2 := (-b - \text{sqrt}(D)) / (2 * a)$



Что неверно?

Документирование программы

- назначение программы
- формат входных данных
- формат выходных данных
- примеры использования программы

Назначение:

программа для решения уравнения

$$ax^2 + bx + c = 0$$

Формат входных данных:

значения коэффициентов a , b и c вводятся с клавиатуры через пробел в одной строке

Документирование программы

Формат выходных данных:

значения вещественных корней уравнения;
если вещественных корней нет, выводится
слово «нет»

Примеры использования программы:

1. Решение уравнения $x^2 - 5x + 6 = 0$

Введите a, b, c: **1 -5 6**

x1=3 x2=2

2. Решение уравнения $x^2 + x + 6 = 0$

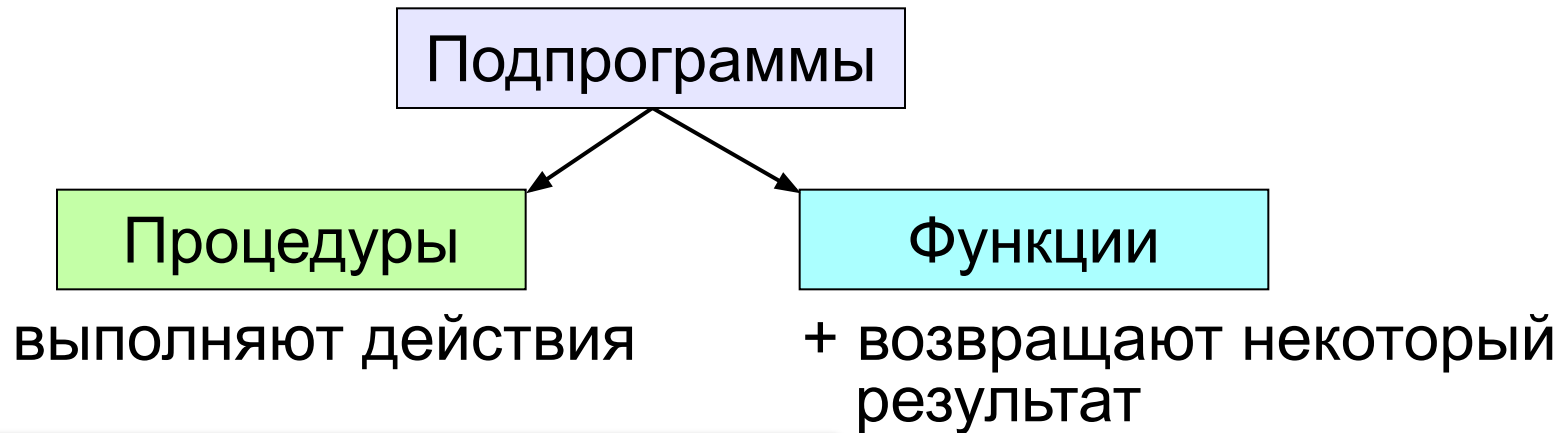
Введите a, b, c: **1 1 6**

Нет.

Программирование (АлгЯзык)

§ 24. Процедуры

Два типа подпрограмм



Процедура или функция?

- а) рисует окружность на экране
- б) определяет площадь круга
- в) вычисляет значение синуса угла
- г) изменяет режим работы программы
- д) возводит число x в степень y
- е) включает двигатель автомобиля
- ж) проверяет оставшееся количество бензина в баке
- з) измеряет высоту полёта самолёта

Простая процедура

```
алг printLine  
нач  
    ВЫВОД '-----', НС  
кон  
...  
кон
```

**ВЫЗОВ
процедуры**



Что делает?



- можно вызывать сколько угодно раз
- нет дублирования кода
- изменять – в одном месте

Линии разной длины

```
алг printLine5
нач
  вывод '-----', нс
кон
```

```
алг printLine10
нач
  вывод '-----'
кон
```

```
алг printLine10
нач
  алг printLine (цел n)
  нач
    цел i
    нц для i от 1 до n
      вывод '-'
    кц
  вывод нс
кон
```

? Как улучшить?

параметр
процедуры

Процедура с параметром

```
алг С процедурой
```

```
нач
```

```
...
```

```
printLine (10)
```

```
...
```

```
printLine (7)
```

```
printLine (5)
```

```
printLine (3)
```

```
кон
```

Аргумент – значение параметра при конкретном вызове.



Что делает?

```
алг printLine (цел n)
```

```
нач
```

```
...
```

```
кон
```

Параметр – величина, от которой зависит работа процедуры.

Несколько параметров

символьная строка

```
алг printLine (лит с, цел n)
нач
  цел i
  нц для i от 1 до n
    вывод с
  кц
  вывод нс
кон
```



Что изменилось?



Как вызывать?

```
printLine ( X, 'X' )
```

```
✓ printLine ( '+', 5 )
```

```
✓ printLine ( '+-+', 5 )
```

В других языках программирования

Паскаль:

```
procedure printLine( n: integer );  
var i: integer;  
begin  
    for i:=1 to n do  
        write( '-' );  
    writeln  
end;
```

В других языках программирования

Python:

```
def printLine (n) :  
    print ('-'*n)
```

C:

```
void printLine (int n)  
{  
    int i;  
    for (i=1; i<=n; i++)  
        putchar ('-');  
}
```


Задачи

«А»: Напишите процедуру, которая принимает параметр – натуральное число N – и выводит на экран две линии из N символов '-'.

Пример:

Длина цепочки: 7

```
-----  
-----
```

«В»: Напишите процедуру, которая принимает один параметр – натуральное число N , – и выводит на экран прямоугольник длиной N и высотой 3 символа.

Пример:

Длина прямоугольника: 7

```
ooooooo  
o      o  
ooooooo
```

Задачи

«С»: Напишите процедуру, которая выводит на экран квадрат со стороной N символов. При запуске программы N нужно ввести с клавиатуры.

Пример:

Сторона квадрата: 5

ooooo

o o

o o

o o

ooooo

Задачи

«D»: Напишите процедуру, которая выводит на экран треугольник со стороной N символов. При запуске программы N нужно ввести с клавиатуры.

Пример:

Сторона : 5

```
o
oo
ooo
oooo
ooooo
```

Рекурсия

Задача. Вывести на экран двоичный код натурального числа.

```
алг printBin (цел n)
нач
    ...
кон
```

Алгоритм перевода через остатки:

```
нц пока n <> 0
    вывод mod (n, 2)
    n := div (n, 2)
кц
```



Что получится при $n = 6$?

011X

в обратном порядке!

Рекурсия

Чтобы вывести двоичную запись числа n , нужно сначала вывести двоичную запись числа $\text{div}(n, 2)$, а затем — его последнюю двоичную цифру, равную $\text{mod}(n, 2)$.

двоичная запись числа 6

110

$\text{mod}(6, 2)$

двоичная запись числа 3



Чтобы решить задачу, нужно решить ту же задачу для меньшего числа!

Это и есть **рекурсия!**



Чтобы понять рекурсию, нужно понять рекурсию! 😊

Рекурсивная процедура

```
алг printBin (цел n)
```

вызывает сама себя!

```
нач
```

```
    printBin (div (n, 2))
```

```
    вывод mod (n, 2)
```

```
кон
```

```
printBin (6)
```



Что получится?

Рекурсивная процедура — это процедура, которая вызывает сама себя.

```
printBin (6)
```

бесконечные вызовы

```
printBin (3)
```

```
printBin (1)
```

```
printBin (0)
```

```
printBin (0)
```



Как исправить?

Рекурсивная процедура

```
алг printBin(цел n)
нач
  если n = 0 то выход все
  printBin(div(n, 2))
  вывод mod(n, 2)
кон
```

printBin(6)



Что получится?

printBin(6)

printBin(3)

printBin(1)

printBin(0)

вывод mod(1, 2)

вывод mod(3, 2)

вывод mod(6, 2)

1 1 0

рекурсия
заканчивается!

Задачи

«**A**»: Напишите рекурсивную процедуру, которая переводит число в восьмеричную систему.

Пример:

Введите число: **66**

В восьмеричной: 102

«**B**»: Напишите рекурсивную процедуру, которая переводит число в любую систему счисления с основанием от 2 до 9.

Пример:

Введите число: **75**

Основание: **6**

В системе с основанием 6: 203

Задачи

«С»: Напишите рекурсивную процедуру, которая переводит число в шестнадцатеричную систему.

Пример:

Введите число: **123**

В шестнадцатеричной: **7B**

«D»: Напишите рекурсивную процедуру, которая переводит число в любую систему счисления с основанием от 2 до 36.

Пример:

Введите число: **350**

Основание: **20**

В системе с основанием 20: **HA**

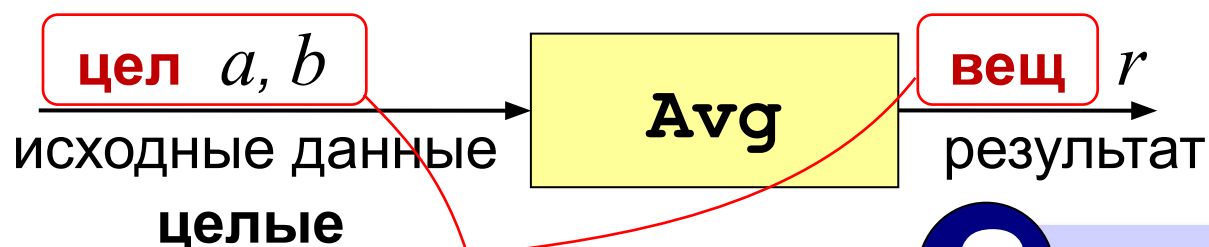
Программирование (АлгЯзык)

§ 25. Функции

Что такое функция?

Функция — это вспомогательный алгоритм, который возвращает результат (число, строку символов и др.).

Задача. Написать функцию, которая вычисляет среднее арифметическое двух целых чисел.



```
алг вещ Avg (цел a, b)  
нач  
  знач := (a+b) / 2  
кон
```

? Тип
результата?

специальная переменная для
записи результата функции

Как вызывать функцию?

Запись результата в переменную:



Чему равно?

```
вещ sr  
sr := Avg(5, 8)
```

6.5

```
цел x=2, y=5  
вещ sr  
sr := Avg(x, 2*y+8)
```

10

Вывод на экран:

```
цел x=2, y=5  
вещ sr  
sr := Avg(x, y+3)  
ВЫВОД Avg(12, 7), нс  
ВЫВОД sr+Avg(x, 12)
```

5

9.5

12

Как вызывать функцию?

Использование в условных операторах:

```
цел a, b  
ввод a, b  
если Avg(a,b) > 5 то  
    вывод 'Да!'  
иначе  
    вывод 'Нет!'  
все
```



Когда печатает «Да»?

Как вызывать функцию?

Использование в циклах:

```
цел а, б  
ввод а, б  
нц пока Avg(a,b) > 0  
    вывод 'Нет!'  
    ввод а, б  
кц  
вывод 'Угадал!'
```



Когда напечатает
«Угадал»?

В других языках программирования

Паскаль:

```
function Avg(a, b: integer): real;  
begin  
    Avg := (a+b) / 2  
end.
```

Python:

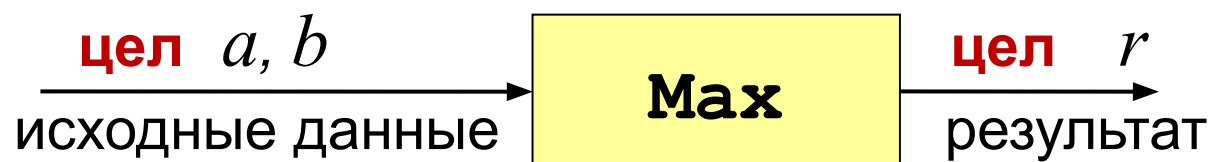
```
def Avg(a, b):  
    return (a+b) / 2
```

C:

```
float Avg(int a, int b)  
{  
    return (a+b) / 2.0;  
}
```

Максимум из двух (трёх) чисел

Задача. Составить функцию, которая определяет наибольшее из двух целых чисел.



```
алг цел Max (цел a, b)
```

```
нач
```

```
    если a>b то
```

```
        знач := a
```

```
    иначе
```

```
        знач := b
```

```
    все
```

```
кон
```



Как с её помощью найти максимум из трёх?

```
алг цел Max3 (цел a, b, c)
```

```
нач
```

```
    знач := Max ( Max (a, b), c )
```

```
кон
```


Сумма цифр числа

Задача. Составить функцию, которая вычисляет сумму значений цифр натурального числа.

```
алг цел sumDigits (цел N0)
```

```
нач
```

```
  цел N, d, sum
```

```
  N := N0
```

| работаем с копией числа

```
  sum := 0
```

| накапливаем сумму с 0

```
нц пока N <> 0
```

```
  d := mod(N, 10)
```

| выделим последнюю цифру

```
  sum := sum + d
```

| добавим к сумме

```
  N := div(N, 10)
```

| удалим последнюю цифру

```
кц
```

```
знач := sum
```

```
кон
```

В КуМире аргумент
нельзя менять!

Задачи

«А»: Напишите функцию, которая вычисляет среднее арифметическое пяти целых чисел.

Пример:

Введите 5 чисел: **1 2 3 4 6**

Среднее: **3.2**

«В»: Напишите функцию, которая находит количество цифр в десятичной записи числа.

Пример:

Введите число: **751**

Количество цифр: **3**

Задачи

«С»: Напишите функцию, которая находит количество единиц в двоичной записи числа.

Пример:

Введите число: **75**

Количество единиц: **4**

Логические функции

Логическая функция — это функция, возвращающая логическое значения (**да** или **нет**).

- можно ли применять операцию?
- успешно ли выполнена операция?
- обладают ли данные каким-то свойством?

Логические функции

Задача. Составить функцию, которая возвращает «да», если она получила чётное число и «нет», если нечётное.

```
алг лог Чётное (цел N)
нач
  если mod(N, 2) = 0 то
    знач := да
  иначе
    знач := нет
все
кон
```

```
алг лог Чётное (цел N)
нач
  знач := (mod(N, 2) = 0)
кон
```

Рекурсивные функции

Рекурсивная функция — это функция, которая вызывает сама себя.

Задача. Составить рекурсивную функцию, которая вычисляет сумму цифр числа.

? Как сформулировать решение рекурсивно?

Сумму цифр числа N нужно выразить через сумму цифр другого (меньшего) числа.

Сумма цифр числа N равна значению последней цифры плюс сумма цифр числа, полученного отбрасыванием последней цифры.

$$\text{sumDig}(12345) = 5 + \text{sumDig}(1234)$$

Рекурсивная функция

Сумма цифр числа N

Вход: натуральное число N.

Шаг 1: $d := \text{mod}(N, 10)$

Шаг 2: $M := \text{div}(N, 10)$

Шаг 3: $s :=$ сумма цифр числа M

Шаг 4: $\text{sum} := s + d$

Результат: sum.

последняя цифра

число без
последней цифры

? Что забыли?

? Когда остановить?

Сумма цифр числа (рекурсия)

```
алг цел sumDigRec (цел N)
```

```
нач
```

```
  цел d, sum
```

```
  если N = 0 то
```

```
    знач := 0
```

```
  иначе
```

```
    d := mod(N, 10)
```

```
    sum := sumDigRec(div(N, 10))
```

```
    знач := sum + d
```

```
  все
```

```
кон
```



Зачем это?



Где рекурсивный вызов?

Задачи

«А»: Напишите логическую функцию, которая возвращает значение «истина», если десятичная запись числа заканчивается на цифру 0 или 1.

Пример:

Введите число: **1230**

Ответ: Да

«В»: Напишите логическую функцию, которая возвращает значение «истина», если переданное ей число помещается в 8-битную ячейку памяти.

Пример:

Введите число: **751**

Ответ: Нет

Задачи

«С»: Напишите логическую функцию, которая возвращает значение «истина», если переданное ей число простое (делится только на само себя и на единицу).

Пример:

Введите число: **17**

Число простое!

Пример:

Введите число: **18**

Число составное!

Конец фильма

ПОЛЯКОВ Константин Юрьевич

д.т.н., учитель информатики

ГБОУ СОШ № 163, г. Санкт-Петербург

kpolyakov@mail.ru

ЕРЕМИН Евгений Александрович

к.ф.-м.н., доцент кафедры мультимедийной

дидактики и ИТО ПГГПУ, г. Пермь

eremin@pspu.ac.ru

Источники иллюстраций

1. иллюстрации художников издательства «Бином»
2. авторские материалы