

Алгоритмы: цикл «N раз»

Компьютер может исполнять миллионы вычислительных операций в секунду, например, повторять одну и ту же последовательность действий. Конечно, программисты не записывают при этом миллион раз одну и ту же последовательность команд, они используют специальную **составную команду алгоритмического языка** (конструкции) — **цикл «N раз»**, где N — целое число:

```
нц <число повторений> раз
    <последовательность команд>
кц
```

Служебные слова **нц** (начало цикла) и **кц** (конец цикла) пишутся строго одно под другим. Повторяемая последовательность команд записывается с отступом вправо.

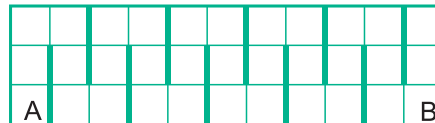
Вот пример алгоритма, в котором используется **цикл «N раз»**. В этом цикле число повторений $N = 5$.

```
алг из А в В
  дано | Робот
        | в клетке А
  надо | Робот
        | в клетке В
нач
  нц 5 раз
    вверх
    вверх
    вправо
    вниз
    вниз
    вправо
  кц
кон
```

При выполнении этого алгоритма компьютер 5 раз повторит последовательность команд

```
вверх
вверх
вправо
вниз
вниз
вправо
```

и *Робот* окажется в клетке В:



Использование цикла позволяет коротко записать длинные последовательности действий. Например, при выполнении алгоритма закрасить ряд из 9 клеток, в котором два раза используется **цикл «N раз»**, компьютер даст *Роботу* 27 команд:

сначала 9 раз будет выполнена пара команд закрасить, вправо, а затем 9 команд влево.

```
алг закрасить ряд из 9 клеток
  дано | на поле Робота нет стен,
        | справа от клетки с Роботом
        | есть 9 клеток поля
  надо | Робот закрасил 9 клеток ряда
        | и вернулся в исходное положение
нач
  нц 9 раз
    закрасить
    вправо
  кц
  нц 9 раз
    влево
  кц
кон
```

Если вместо числа 9 поставить число 100, *Робот* выполнит уже 300 команд, но при этом количество строк в алгоритме не увеличится! Таким образом, с помощью составной команды **цикл «N раз»** можно коротко записывать алгоритмы, которые описывают очень длинные последовательности действий.

Внутри цикла, как и в других местах алгоритма, можно вызывать вспомогательные алгоритмы — смотри, например, алгоритм закрасить квадрат 1 на следующей странице.

Внутри цикла могут быть и другие циклы — смотри, например, алгоритм закрасить квадрат 2. Цикл, который находится внутри другого, называется **вложенным циклом**. В данном случае использование вложенных циклов не очень удобно: алгоритм закрасить квадрат 2 выглядит более громоздким и менее понятным, чем алгоритм закрасить квадрат 1.

Правила алгоритмического языка допускают задание любого целого числа повторений N в конструкции **цикл «N раз»**. Это число может равняться нулю и даже быть отрицательным! Это не

```

алг закрасить квадрат 1
  дано | поле Робота без стен,
        | справа от клетки с Роботом есть 9 клеток поля,
        | сверху над клеткой с Роботом есть 9 клеток поля
  надо | Робот закрасил квадрат  $9 \times 9$  и
        | вернулся в исходное положение

нач
  нц 9 раз
    закрасить ряд из 9 клеток
    вверх
  кц
  нц 9 раз
    вниз
  кц
кон

```

```

алг закрасить квадрат 2
  дано | поле Робота без стен,
        | справа от клетки с Роботом есть 9 клеток поля,
        | сверху над клеткой с Роботом есть 9 клеток поля
  надо | Робот закрасил квадрат  $9 \times 9$  и
        | вернулся в исходное положение

нач
  нц 9 раз
    нц 9 раз
      закрасить
      вправо
    кц
    нц 9 раз
      влево
    кц
    вверх
  кц
  нц 9 раз
    вниз
  кц
кон

```

считается ошибкой и не приведёт к отказу при выполнении алгоритма. В таком случае тело цикла не будет выполнено ни разу, а компьютер перейдёт к командам, записанным после **кц**.

66

Составь алгоритм для *Робота*, который закрашивает все клетки на квадратном поле без внутренних стен размером 10×10 . В начальном состоянии *Робот* находится в левом верхнем углу поля, после выполнения алгоритма *Робот* оказывается в правом нижнем углу поля. Используй как вспомогательный алгоритм закрасить ряд из 9 клеток со с. 44 и составную команду **цикл «N раз»**.

67

Нарисуй результат выполнения *Роботом* алгоритма *узор2*. Сколько раз *Робот* выполнил команду закрасить, выполняя алгоритм *узор2*? Сколько всего на поле стало закрашенных клеток? Сколько клеток *Робот* закрасил по одному разу?

```
алг узор2
  дано | поле Робота без стен, 10 × 10 клеток,
       | Робот в верхнем левом углу поля
  надо |
  нач
    нц 5 раз
      нц 3 раз
        закрасить
        вправо
      кц
    нц 3 раз
      закрасить
      вниз
    кц
  нц 2 раз
    закрасить
    влево
    вверх
    закрасить
  кц
кц
кон
```

68

Выписаны подряд без пробелов все натуральные числа от 1 до 50 включительно. Какая цифра стоит на 79-м месте?

69

Заполни таблицу выполнения *Кузнечиком* с системой команд вперёд 13, назад 8 алгоритма следующее число по образцу.

алг следующее число
дано |
надо | Кузнечик переместился
 | на шаг вправо
нач
нц 5 **раз**
 вперёд 13
кц
нц 8 **раз**
 назад 8
кц
кон

Команда	Результат
вперёд 13	13
вперёд 13	26
...	

70

Составь алгоритм, который, как и алгоритм следующее число, переводит *Кузнечика* с системой команд вперёд 13, назад 8 на шаг вправо, но так, что при его выполнении *Кузнечик* не удаляется от начального положения больше чем на 50 шагов. Используй составную команду **цикл «N раз»**.

71

Даны три множества латинских букв. Построй:

- пересечение множеств А и Б;
- объединение множеств Б и В;
- пересечение множеств Б и В;
- пересечение множеств А, Б и В.

А

Q W E
 T R S
 Y U F
 G J L

Б

Z W V
 N A D
 Y U S
 I H P

В

I U D
 W H S
 A V P

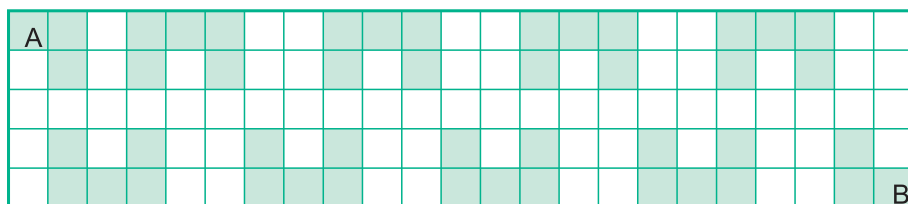
72 Составь алгоритм для *Кузнечика* с системой команд *вперёд 3*, *назад 2*, который переводит *Кузнечика* из точки «0» в точку «360». Используй составную команду **цикл «N раз»**.

73 Составь алгоритм для *Кузнечика* с системой команд *вперёд 3*, *назад 2*, который заставит *Кузнечика* перейти из точки «0» в точку «100», побывав при этом в каждой точке от 0 до 100 ровно по одному разу. Используй составную команду **цикл «N раз»**.



Прежде чем решить эту задачу, реши аналогичную задачу для короткого отрезка, например для отрезка от 0 до 5.

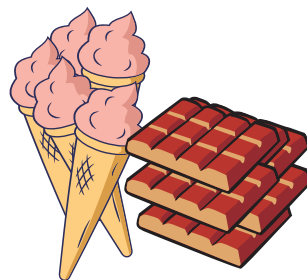
74 Составь алгоритм, который переводит *Робота* из точки А в точку В и раскрашивает все те клетки, которые раскрашены на поле на рисунке. Используй составную команду **цикл «N раз»** и вложенный цикл.



75 Составь другой алгоритм, который тоже переводит *Робота* из точки А в точку В и раскрашивает все те клетки, которые раскрашены на поле на рисунке (см. рисунок из задачи 74), но в котором используется не вложенный цикл, а вспомогательные алгоритмы.

76 Реши задачу, построив дерево перебора вариантов.

В школьной столовой продают шоколадки по 15 р. и мороженое по 12 р. за порцию. У Оли есть 102 р. Сколько Оле нужно купить шоколадок и сколько порций мороженого, чтобы истратить все свои деньги? Найди все возможные варианты такой покупки.

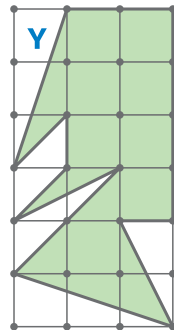


77

Построй алгоритм для *Удвоителя* (система команд *Удвоителя* приведена на с. 22), после выполнения которого на экране отобразится число 2014. Используй только составную команду **цикл «N раз»** и команду **прибавь 1**. Дай этому алгоритму имя **прибавь 2014 раз по 1**. Сколько всего команд компьютер передаст *Удвоителю* при выполнении алгоритма **прибавь 2014 раз по 1**? Затем составь такой алгоритм, после выполнения которого на экране тоже отобразится число 2014, но при выполнении которого компьютер передаст *Удвоителю* не больше 20 команд. Дай этому алгоритму имя **получи 2014**.

78

Нарисуй, как разрезать многоугольник Y, чтобы получились два многоугольника на сетке одинаковой площади.



79

Сколько было брёвен, если, сделав 52 распила (каждый раз распиливали ровно одно бревно), из них получили 72 полена?

80

Найди все возможные варианты чисел, которые могут быть отображены на экране *Удвоителя* после выполнения им не больше 5 команд из начального положения «1». Для решения построй дерево перебора вариантов. Пользуясь построенным деревом, найди все числа второго десятка, которые нельзя получить на экране *Удвоителя* не больше чем за 5 команд.

81

Вот греческие буквы. Найди три одинаковые буквы, напиши такую букву в тетради. Найди букву, которая встречается здесь ровно один раз, напиши такую букву в тетради.

θ ω ρ τ ψ υ ι ο π ε σ δ φ
 γ η φ κ λ ζ ξ χ β α ν μ θ
 ω ε π ρ ο τ ι ψ υ ξ σ δ λ
 φ κ γ φ η ζ χ π μ ν β