

- C** В последовательности R жёлтая круглая бусина идёт раньше синей квадратной.
- D** В последовательности R предыдущая бусина перед каждой круглой — треугольная.
- E** В последовательности R каждая квадратная бусина — красная.
- F** В последовательности R вторая бусина после каждой квадратной — круглая.
- G** В последовательности R вторая бусина после квадратной — круглая.
- H** В последовательности R десятая с конца бусина — красная.
- J** В последовательности R предыдущая бусина перед жёлтой треугольной — красная.

Алгоритм

Для программного управления нужно заранее описать последовательность команд, которые нужно дать исполнителю. Причём это описание должно быть понятно компьютеру, а значит, написано на специальном языке. Такое описание последовательности команд мы будем называть **алгоритмом**.

На школьном алгоритмическом языке алгоритм записывается так:

Заголовок алгоритма	{	алг имя алгоритма дано условия применения алгоритма надо цель выполнения алгоритма
Тело алгоритма	{	нач <последовательность команд> кон

Например, для задачи «Ход конём» со с. 94 алгоритм будет выглядеть так:

```
алг ход конём
  дано | Робот в клетке А, стен
        | на поле нет
  надо | Робот в клетке В
нач
  вправо
  вправо
  вниз
кон
```

Слова **алг** (сокращение от «алгоритм»), **дано**, **надо**, **нач** (сокращение от «начало»), **кон** (сокращение от «конец») называются *служебными словами*, они нужны для оформления алгоритма. Служебные слова **алг**, **нач** и **кон** пишутся строго одно под другим. Команды между служебными словами **нач** и **кон** пишутся с отступом — так алгоритм легче читать.

Строки **алг**, **дано** и **надо** образуют *заголовок алгоритма*. *Имя алгоритма* может состоять из одного или нескольких слов. Удобно подбирать такое имя, чтобы можно было понять, для чего предназначен алгоритм. В строке **дано** описываются свойства начального состояния, которые считаются известными, в строке **надо** — свойства, которые должны быть у состояния исполнителя после выполнения им алгоритма.

Часть алгоритма, заключённая между словами **нач** и **кон**, называется *телом* алгоритма. Тело содержит последовательность команд — программу для исполнителя.

Комментарии в алгоритмическом языке.

В алгоритме `ход конём` после знака `|` в строках **дано** и **надо** записан комментарий. Такие комментарии разрешается помещать в конце любой строки, отделяя их знаком `|`. Если комментарий занимает несколько строк, то знак `|` перед комментарием надо писать в каждой строке. Комментарии могут записываться в любой удобной человеку форме: при выполнении алгоритма компьютер полностью пропускает комментарии — алгоритм



выполняется так же, как если бы комментариев вообще не было. Таким образом, комментарии предназначены исключительно для человека — они облегчают ему понимание алгоритма.

Исполнение алгоритма. Получив приказ исполнить алгоритм, компьютер выполняет следующие действия:

1. Проверяет, соблюдаются ли начальные условия, указанные в строке **дано** (пока в **дано** пишем только комментарии, но позднее мы научимся писать там условия, которые компьютер сможет проверять сам).
2. Последовательно читает команды после строки **нач** и передаёт их исполнителю.
3. Встретив строку **кон**, проверяет достижение цели алгоритма, записанной в строке **надо** (в **надо** пока пишем только комментарии, но позднее мы научимся писать там условия, которые компьютер сможет проверять сам).
4. Заканчивает выполнение алгоритма.

Пример. Компьютер получает приказ исполнить алгоритм `ход конём`:

1. В строке **дано** записан только комментарий — компьютер его пропускает.

```
алг  ход конём
     дано | Робот в клетке А, стен
           | на поле нет
     надо | Робот в клетке В

     нач
         вправо
         вправо
         вниз
     кон
```

2. Компьютер последовательно даёт команды *Роботу*: вправо, вправо, вниз. *Робот* исполняет эти команды.
3. Дойдя до строки **кон**, компьютер проверяет, что написано в строке **надо**. Там записан только комментарий — компьютер его пропускает.
4. Компьютер заканчивает выполнение алгоритма `ход конём`.

Ошибки в алгоритмах

Если при составлении алгоритма мы случайно допустим синтаксическую ошибку, например, вместо `вниз` напишем `внис` или вместо `вправо` напишем `направо`, то компьютер нашу запись не поймёт и, не приступая к выполнению алгоритма, сообщит об ошибке.

Но даже если все команды в алгоритме записаны правильно, это ещё не значит, что алгоритм составлен без ошибок — исполнитель, выполняя его, делает именно то, что мы задумали. Ошибки в составлении алгоритма называются *логическими*.

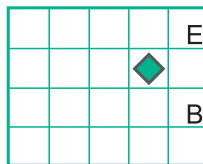
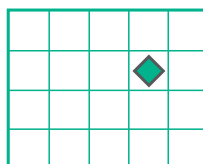
Иногда логическая ошибка может привести к *отказу* — невозможности выполнить очередную команду. Например, при попытке выполнить алгоритм `ход конём` из состояния, изображённого на рисунке справа, компьютер попытается последовательно вызвать команды `вправо`, `вправо`, `вниз`.

Однако вторую команду `вправо` *Робот* выполнить не сможет, и возникнет *отказ*. Получив от исполнителя сигнал отказа, компьютер сообщит об ошибке и прекратит выполнение алгоритма.

У каждого исполнителя свои причины отказов. Например, отказ *Робота* возникает при попытке идти сквозь стену — внутреннюю стену или границу поля.

Ошибки в алгоритме не всегда приводят к отказам. В алгоритме могут быть логические ошибки, которые компьютер не может обнаружить ни до выполнения алгоритма, ни при его выполнении. Например, если в алгоритме `ход конём` мы вместо команды `вниз` случайно напишем `вверх`, то компьютер выполнит алгоритм, но *Робот* из начальной позиции (см. рисунок справа) переместится не в клетку В, а в клетку Е, но при этом никаких сообщений об ошибках мы не получим. Да и откуда компьютеру знать, куда мы на самом деле хотели переместить *Робота*, ведь в строке `надо` записан только комментарий (а *Робот* комментарий не читает)!

В правильно составленных алгоритмах, конечно, никаких ошибок быть не должно. Но если синтаксические ошибки устранить легко, то поиск и устранение логических ошибок могут оказаться весьма трудным делом.



191

Даны алгоритмы А и Б, в которых стёрты комментарии и названия.

А алг _____
 дано | _____
 надо | _____
 нач
 вверх
 закрасить
 вниз
 вправо
 закрасить
 влево
 вниз
 закрасить
 вверх
 влево
 закрасить
 вправо
КОН

Б алг _____
 дано | _____
 надо | _____
 нач
 вверх
 вправо
 закрасить
 вниз
 вниз
 закрасить
 влево
 влево
 закрасить
 вверх
 вверх
 закрасить
 вправо
 вниз
КОН

Опиши движение *Робота* в процессе выполнения каждого алгоритма. Используй для этого клетки тетради, считая, что весь лист тетради — это поле *Робота*. Постарайся выбрать такое начальное положение *Робота*, чтобы при выполнении алгоритма стенки на его пути не встретились — не возникло отказа. Отметь крестиком клетку, выбранную тобой для начального положения *Робота*, закрась клетки, которые *Робот* закрасит в результате выполнения алгоритма, отметь кружком (ноликом) клетку, в которой *Робот* окажется после выполнения программы. Придумай и напиши в тетради подходящее название алгоритма и комментарии после слов **дано** и **надо**.

192

Измени алгоритм Б задачи 191 так, чтобы при его исполнении *Робот*:

- прошёл тем же маршрутом, но ничего не закрашивал;
- закрасил все клетки, в которых он побывал.

Напиши получившиеся программы. Опиши движение *Робота* в процессе выполнения каждого алгоритма так, как указано в задаче 191.

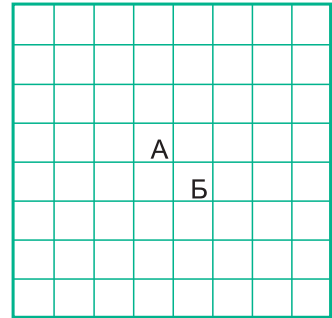
193

Исполнитель *Квадратор* похож на исполнителя *Удвоитель*: у него тоже есть экран, на котором отображается одно число, и есть две кнопки. Система команд исполнителя *Квадратор* состоит из двух команд: прибавь 1 и возведи в квадрат. В начальном состоянии на экране *Квадратора* отображается число 1. Напиши программу для *Квадратора*, после выполнения которой на экране будет отображаться число: а) 5; б) 12; в) 25. Постарайся, чтобы команд в каждой программе было как можно меньше: обсуди с другими ребятами в классе, у кого получилось меньше всего команд.

194

Составь алгоритм, при выполнении которого *Робот* переместится из клетки А в клетку Б. Существует ли такой алгоритм, переводящий *Робота* из клетки А в клетку Б, при выполнении которого *Робот* делает:

- а) два шага;
- б) четыре шага;
- в) семь шагов;
- г) 2014 шагов?

**195**

Пусть на поле *Робота* нет стен и закрашенных клеток. Не делая рисунка, определи, сколько клеток будет закрашено после исполнения следующих алгоритмов (считается, что *Робот* в начальном положении находится далеко от границ поля и при выполнении этих программ отказа не возникнет):

алг А
дано |
надо |
нач
 закрасить
 вправо
 вверх
 закрасить
 вправо
 закрасить
 вверх
 закрасить
 закрасить
 вправо
кон

алг Б
дано |
надо |
нач
 закрасить
 вправо
 закрасить
 закрасить
 вправо
 вправо
 закрасить
 закрасить
 закрасить
 вправо
кон

196

Петя составил алгоритм, а Коля стёр в нём одну команду. Определи, какую команду стёр Коля, если известно, что при выполнении составленного Петей алгоритма *Робот* возвращался в исходное положение.

```

алг А
  дано | стен на поле нет
  надо | Робот погулял и вернулся
        | в исходное положение

нач
  вверх
  вправо

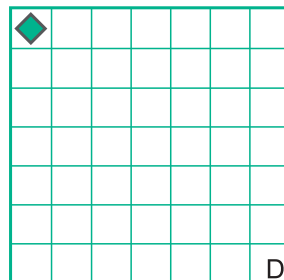
  _____
  вниз
  влево
  влево
кон
  
```

197

Антон составил алгоритм, при выполнении которого *Робот* вернулся в исходное положение. Коля стёр одну из команд. При выполнении Колиного алгоритма *Робот* также вернулся в исходное положение. Какую команду стёр Коля?

198

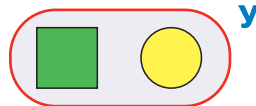
Сима составила программу, которая переводит *Робота* в точку D (поле и начальное положение *Робота* изображены справа). Она написала каждую команду на отдельной карточке и сложила их стопкой в нужном порядке. Но только Сима собралась переписать программу в тетрадь, как прибежал Ваня и переложил карточки в другом порядке. Как ни странно, после этого тоже получилась программа, которая переводит *Робота* в клетку D. Объясни, как это получилось.

**199**

Построй четыре разные последовательности бусин, для каждой из которых следующие утверждения истинны:

Длина этой последовательности — 2.

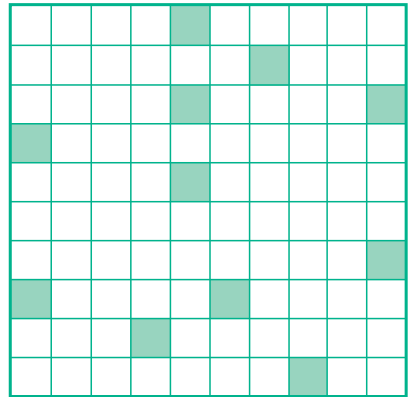
Каждая бусина этой последовательности есть в множестве $У$.



200

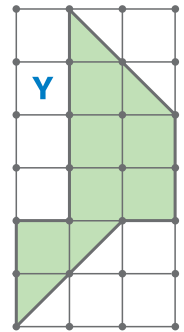


Начав с верхнего левого угла поля, *Робот* прошёл в нижний правый угол поля, побывав в каждой незакрашенной клетке поля ровно по одному разу, а в закрашенных клетках ни одного раза. При этом сам *Робот* никаких клеток не закрашивал. Каким мог быть путь *Робота*? Нарисуй такое же поле в тетради и проведи линию пути *Робота*.



201

Придумай, как разрезать многоугольник *У* на два одинаковых многоугольника на сетке. Начерти такую же фигуру по клеткам тетради, нарисуй линии разреза.



202

Среди данных множеств найди:

- а) множество, равное пересечению множества всех треугольных бусин и множества всех красных бусин;
- б) множество, равное объединению множества всех синих бусин и множества всех круглых бусин;
- в) множество, равное пересечению множества всех бусин и множества всех синих бусин;
- г) множество, равное объединению множества всех жёлтых бусин и пустого множества.

В ответе напиши имена множеств.



А
Б
В
Г

Д
Е
Ж

З
И