

Двое играют в следующую игру: каждый игрок по очереди вычёркивает одно число из ряда **1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19** до тех пор, пока не останется два числа. Если сумма этих чисел делится на 5, то выигрывает первый игрок, если не делится — второй. У кого из игроков в этой игре есть выигрышная стратегия? Опиши эту стратегию.

## Составные условия: слова «И», «ИЛИ», «НЕ»

При составлении алгоритмов иногда бывает нужно использовать не одно условие, а сразу два.

Пусть, например, *Робот* стоит на поле, в котором нет внутренних стен. Чтобы понять, находится ли *Робот* в верхнем левом углу поля, нужно выяснить истинность двух условий: слева стена, сверху стена. Если оба условия *одновременно истинны*, то *Робот* стоит в левом верхнем углу поля (на котором нет внутренних стен). Если *хотя бы одно из этих условий ложно*, то *Робот* стоит в другой клетке поля.

Другой пример. Нам надо выяснить, не находится ли *Робот* рядом с любой из двух вертикальных границ поля. Для этого мы должны проверить истинность двух условий: слева стена, справа стена. Если *хотя бы одно из этих условий истинно*, то *Робот* находится рядом с одной из вертикальных границ поля. Если *оба условия одновременно ложны*, то *Робот* не находится у вертикальной стены.

Посмотри: чтобы объяснить, какие значения истинности должны иметь два условия, нам пришлось написать целый абзац. Чтобы иметь возможность это записать кратко, в информатике и в математике принято использовать слова «и», «или», «не». При этом условие, которое получается из простых условий с помощью слов «и», «или», «не», называется **составным условием**.

**Слово «и».** В русском языке сложное предложение, образованное с помощью союза «и» (или с помощью близкого по значению союза «а»), *истинно в том и только в том случае, когда истинны оба составляющие его простые предложения*

(утверждения). Например, предложение «Лондон — столица Англии, а Париж — столица Франции» истинно, а каждое из следующих предложений ложно:

«Лондон — столица Англии, а Париж — столица Норвегии»,

«Лондон — столица Швеции, а Париж — столица Франции»,

«Лондон — столица Швеции, а Париж — столица Норвегии».

В информатике и в математике значение истинности утверждения, состоящего из двух утверждений (условий), связанных словом «и», определяется точно так же, как в русском языке.

Это можно записать с помощью такой таблицы истинности (здесь А, В — имена двух простых утверждений):

А	В	А и В
И	И	И
И	Л	Л
Л	И	Л
Л	Л	Л

Из таблицы видно: если оба условия А и В истинны, то истинно и составное условие «А и В». Если хотя бы одно из простых условий ложно (или ложны оба), то составное условие ложно. Таким образом, если *Робот* находится в левом верхнем углу поля, то он на составное условие

слева стена **и** сверху стена

ответит «да». Если *Робот* находится в другой клетке поля без внутренних стен, то *Робот* ответит «нет», так как либо сверху, либо слева от него стены не будет.

**Слово «или».** В русском языке союз «или» имеет несколько разных значений. В математике используется только одно значение — такое, как, например, в предложении «Здесь близко река или озеро», иначе «Здесь близко река или здесь близко озеро». Это предложение образовано из двух простых предложений с помощью союза «или». При этом говорящий не знает точно, истинно ли каждое из простых предложений, но утверждает, что *хотя бы одно из них истинно, а может быть, и оба*. Это предложение

будет истинным, если близко есть только река, но нет озера, или реки нет, но озеро есть, или есть и озеро, и река. Ложным это предложение будет, только если близко нет ни реки, ни озера.

Это можно записать с помощью такой таблицы истинности:

А	В	А или В
И	И	И
И	Л	И
Л	И	И
Л	Л	Л

Таким образом, если *Робот* находится около какой-нибудь из вертикальных границ поля, то он на составное условие слева стена **или** справа стена

ответит «да». Если *Робот* находится в другой клетке поля без внутренних стен, то *Робот* ответит «нет», так как ни справа, ни слева от него стены не будет.

**Слово «не».** В русском языке частица «не» имеет несколько значений. Но если в предложение добавить частицу «не» перед сказуемым, то в новом предложении будет отрицаться то, что утверждалось в старом. Например: «Он не ездил вчера в город», «Кит — это не рыба». В таком случае полученное предложение называется **отрицанием** исходного.

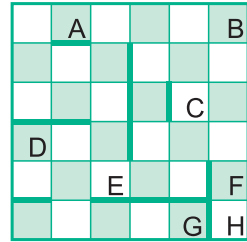
В информатике и в математике слово «не» используется именно в этом значении — при приписывании «не» к утверждению новое утверждение получает противоположное значение истинности: если утверждение А истинно, то утверждение не А ложно, и наоборот. Это можно записать с помощью такой таблицы истинности:

А	не А
И	Л
Л	И

В школьном алгоритмическом языке отрицание условия тоже используется. При этом полученное условие, например **не** сверху стена, мы тоже будем называть **составным**. Составное условие **не** сверху стена имеет противоположное значение истинности по сравнению с простым условием сверху стена.

171

Определи истинность каждого составного условия в таблице для каждой из отмеченных клеток поля. Заполни такую таблицу в тетради — напиши в каждой клетке букву И или Л.



Составное условие	A	B	C	D	E	F	G	H
сверху свободно <b>или</b> снизу свободно								
слева свободно <b>или</b> сверху стена								
справа стена <b>и</b> клетка закрашена								
клетка чистая <b>и</b> снизу свободно								
<b>не</b> слева стена								
<b>не</b> снизу свободно								

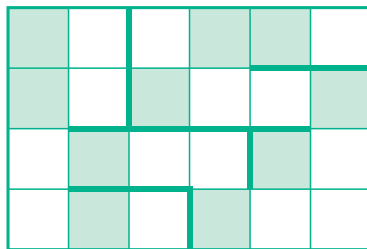
172

Для каждого составного условия с частицей «не» напиши простое условие с таким же значением истинности — построй и заполни таблицу в тетради.

№	Составное условие	Простое условие
1	<b>не</b> клетка закрашена	
2	<b>не</b> справа свободно	
3	<b>не</b> снизу стена	
4	<b>не</b> слева стена	
5	<b>не</b> сверху свободно	
6	<b>не</b> справа стена	
7	<b>не</b> клетка чистая	

173

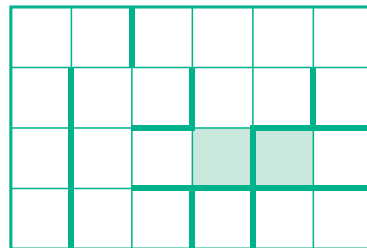
Нарисуй такое же поле в тетради, напиши в каждой клетке номер составного условия, которое истинно для этой клетки. При этом, конечно, в клетке может стоять несколько номеров.



№	Составное условие
1	клетка закрашена <b>или</b> сверху свободно
2	снизу свободно <b>и</b> слева свободно
3	клетка чистая <b>и</b> слева стена
4	<b>не</b> клетка закрашена
5	<b>не</b> справа свободно

174

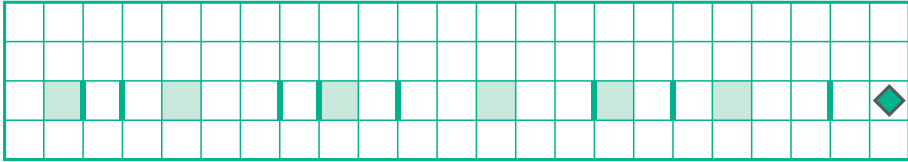
Пользуясь заполненной таблицей, для каждого имени найди какую-нибудь клетку на поле, которой соответствуют указанные значения истинности составных условий. Нарисуй такое поле в тетради, расставь имена клеток так, чтобы таблица была верной.



Составное условие \ Имя клетки	Имя клетки							
	A	B	C	D	E	F	G	H
клетка закрашена <b>или</b> сверху стена	И	И	И	И	Л	Л	Л	Л
снизу свободно <b>или</b> слева свободно	И	Л	Л	И	И	Л	Л	И
справа стена <b>и</b> снизу стена	Л	Л	И	И	Л	Л	И	И

175

Нарисуй состояние *Робота* после выполнения алгоритма бег с препятствиями из данного начального состояния.



```

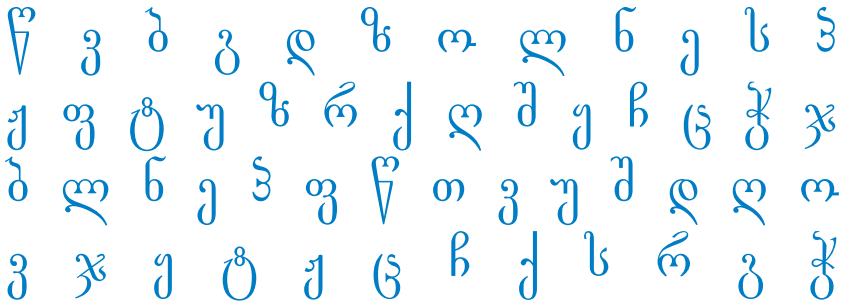
алг бег с препятствиями
  дано |
  надо |
  нач
  нц 22 раз
    если слева свободно и клетка чистая
      то закрасить
        влево
      иначе вверх
        влево
        вниз
    все
  кц
кон
  
```

176

Найди все возможные числа, в которые может попасть *Кузнецик* с системой команд вперёд 3, назад 2, после выполнения им не больше 15 команд из начального положения «100».

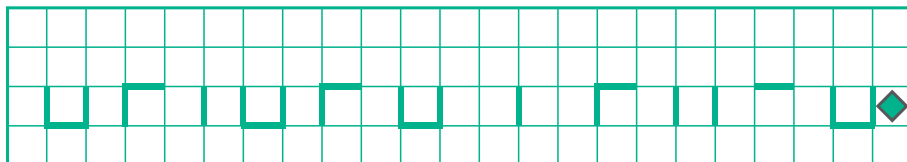
177

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



178

Допиши алгоритм бег с препятствиями 2 — напиши составное условие с союзом «и» так, чтобы при выполнении этого алгоритма из данного начального состояния не возникло отказа. Нарисуй состояние *Робота* после выполнения этого алгоритма из данного начального состояния.



```

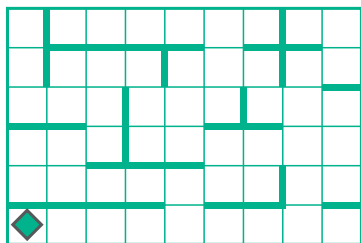
алг бег с препятствиями 2
  дано |
  надо |
  нач
  нц 22 раза
    если _____ и _____
      то вверх
        влево
        вниз
      иначе вниз
        влево
        вверх
    все
  кц
кон

```

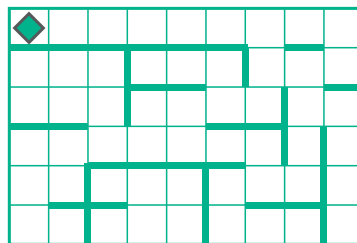
179

Нарисуй состояние *Робота* после выполнения алгоритма выход из коридора из каждого из двух данных начальных состояний. Построй таблицу по образцу из задачи 108 — в каждую клетку налей заготовку поля, раскрась клетки, нарисуй положение *Робота*.

Начальное состояние 1:



Начальное состояние 2:



```

алг выход из коридора
  дано |
  надо |
нач
  нц 7 раз
    если снизу стена и сверху стена
      то вправо
        закрасить
    все
  кц
  если снизу свободно
    то нц 9 раз
      если снизу свободно
        то вниз
          закрасить
      иначе влево
        закрасить
    все
  кц
  иначе нц 9 раз
    если сверху свободно
      то вверх
        закрасить
    иначе вправо
      закрасить
  все
  кц
все
кон

```

180

*Робот* выполнил алгоритм выход из коридора из задачи 179 на поле  $7 \times 7$  клеток. Нарисуй поле и расставь в нём внутренние стенки так, чтобы в результате выполнения *Роботом* этого алгоритма на твоём поле *Робот*, начав из нижнего левого угла, оказался в верхнем правом углу поля.

181

*Робот* находится внутри прямого коридора шириной в 1 клетку, открытого с двух концов, неизвестной длины, идущего в неизвестном направлении. В коридоре есть боковые проходы, ведущие в тупики, при этом нет ни одной клетки, из которой есть сразу два боковых выхода. Составь алгоритм, выводящий *Робота* из этого коридора.