



**А. Л. СЕМЁНОВ  
Т. А. РУДЧЕНКО**

# ИНФОРМАТИКА



# 6

0001011010010

**А. Л. СЕМЁНОВ  
Т. А. РУДЧЕНКО**

# **ИНФОРМАТИКА**



**6** **КЛАСС**

**Учебник**

Допущено Министерством просвещения  
Российской Федерации

2-е издание, стереотипное

Москва «Просвещение» 2022



УДК 373.167.1:004+004(075.3)  
ББК 32.81я721  
С30

На учебник получены **положительные** заключения  
**научной** (заключение РАО № 451 от 14.11.2016 г.),  
**педагогической** (заключение РАО № 138 от 05.10.2016 г.)  
и **общественной** (заключение РКС № 128-ОЭ от 19.12.2016 г.)  
экспертиз.

Издание разработано при поддержке Института киберне-  
тики и образовательной информатики Федерального исследо-  
вательского центра «Информатика и управление» Российской  
академии наук.

Уроки, посвящённые исполнителям, написаны в сотрудни-  
честве с *А. Г. Кушниренко* и *М. А. Ройтбергом*.

Уроки, посвящённые биоинформатике, написаны  
*М. А. Ройтбергом* совместно с *Т. А. Рудченко* и *Е. С. Архиповой*.

Авторы благодарны профессору *А. В. Гладкому* и его книге  
«Введение в современную логику».

Компьютерный проект разработан *Е. Н. Хохловой*.

Издание выходит в pdf-формате.

**Семёнов, Алексей Львович.**

С30 Информатика : 6-й класс : учебник : издание в pdf-формате /  
А. Л. Семёнов, Т. А. Рудченко. — 2-е изд., стер. — Москва :  
Просвещение, 2022. — 159, [1] с. : ил.

ISBN 978-5-09-101745-8 (электр. изд.). — Текст : электрон-  
ный.

ISBN 978-5-09-093739-9 (печ. изд.).

Учебно-методический комплект для 5—6 классов состоит из учебника, тетради  
проектов и пособия для учителя, которое содержит сведения о построении всего  
курса информатики, тематическое планирование, комментарии важных понятий  
курса, обсуждение и решение задач, подробные инструкции по работе в проектах  
и в компьютерном практикуме и др.

Среду Кумир для компьютерного практикума можно скачать с сайта разработчи-  
ков <http://www.niisi.ru/kumir> Электронная версия пособия для учителя размещена  
на сайтах: [www.int-edu.ru](http://www.int-edu.ru) и [www.prosv.ru](http://www.prosv.ru)

УДК 373.167.1:004+004(075.3)  
ББК 32.81я721

ISBN 978-5-09-101745-8 (электр. изд.)  
ISBN 978-5-09-093739-9 (печ. изд.)

© АО «Издательство «Просвещение», 2019  
© Художественное оформление.  
АО «Издательство «Просвещение», 2019  
Все права защищены

# Введение

Дорогой друг!

Мы продолжаем наш курс информатики. В этом году, как и в прошлом, ты будешь строить алгоритмы для управления исполнителями, познакомишься с новыми важными конструкциями алгоритмического языка. Кроме этого, мы обсудим разные виды сортировок, графы и деревья перебора, различные математические игры и выигрышные стратегии в таких играх. Отдельный раздел учебника посвящён шифрованию и биоинформатике.

Как и в прошлом году, каждая глава начинается с объяснительного текста. Рассмотрев и прочитав его, ты сможешь самостоятельно ознакомиться с новыми понятиями, которые будешь использовать для решения задач — самой важной части учебника. Задачи в учебнике встречаются очень разные. Есть задачи простые и сложные, на сообразительность и смекалку, есть задачи математические, лингвистические, биологические, географические. Как ты уже знаешь, все эти области знания сегодня обращаются к информатике.

Задачи с синим номером являются необязательными, но это не значит, что решать их не стоит, ведь самые интересные и увлекательные задачи находятся именно среди них. Для решения некоторых задач удобно сначала вырезать заготовку (например, таблицы) из листа вырезания тетради проектов, наклеить её в тетрадь и заполнить. Такие задачи помечены значком «ножницы».

Компьютерный практикум в этом году будет в основном посвящён работе с исполнителями. Помимо этого, мы предлагаем тебе научиться снимать и монтировать видео — в проекте «С видеокамерой в руках...» у тебя будет возможность поучаствовать в создании небольшого фильма.

Желаем успехов!

**Обозначения:**



Обязательная задача.



Необязательная задача.



Важная информация.



Указание.



К задаче есть лист вырезания в тетради проектов.



# Сортировка: упорядочение и классификация

В информатике сортировкой называется наведение порядка в некотором наборе объектов. Представьте, как долго мы будем искать нужную книгу в библиотеке, если книги будут там просто свалены в кучу на полу. Поэтому в библиотеке книги *упорядочены* — каждая книга находится на строго определённом месте. Упорядочено и расписание движения поездов, самолётов или автобусов. Задачи составления расписаний и вообще упорядочивания большого количества объектов (в информатике говорят «больших массивов») решаются сегодня при помощи компьютера.

Если мы хотим, чтобы в некотором наборе объектов можно было легко найти нужный, мы должны разложить эти объекты в определённом порядке — *рассортировать*. Это можно сделать по-разному.

Один из *способов сортировки* — расположить все объекты в последовательность по определённому, удобному для нас правилу. Такой процесс сортировки называется **упорядочением**.

Например, при составлении словарей обычно слова располагают в *словарном (лексикографическом) порядке*. Общее правило словарного порядка выглядит так:

1. Сравним первые буквы двух слов: если эти буквы разные, то раньше будет идти то слово, первая буква которого идёт раньше в алфавите.
2. Если у двух слов первые буквы одинаковые, то сравним вторые буквы: если эти буквы разные, то раньше будет идти то слово, вторая буква которого идёт раньше в алфавите.
3. Если и вторые буквы у двух слов одинаковые, то сравним третьи буквы: если эти буквы разные, то раньше будет идти то слово, третья буква которого идёт раньше в алфавите.
4. Если и третьи буквы у двух слов одинаковые, будем сравнивать четвёртые, пятые и т. д., пока не дойдём до двух разных букв или пока одно из слов не закончится. В этом случае раньше идёт то слово, которое короче.

Есть словари, в которых слова упорядочены по другим правилам, об этом мы поговорим потом.

Можно сортировать объекты и по-другому — *классифицировать* их по группам по определённому правилу. Например, при сортировке пуговиц у бабушки в комоде достаточно разложить их по кучкам, скажем, так, чтобы в каждой кучке все пуговицы были одинаковыми, или так, чтобы в каждой кучке все пуговицы были одного размера, или так, чтобы в каждой кучке все пуговицы были одного тона.

Такая сортировка называется **классификацией**. В результате классификации получается несколько групп объектов, а внутри каждой группы объекты не упорядочены.

Иногда при сортировке бывает удобно использовать и классификацию, и упорядочение: сначала классифицировать объекты, а затем внутри каждой группы их упорядочивать. Так, например, часто сортируют книги в библиотеке. Сначала классифицируют по теме: приключения, фантастика, книги о животных и др., а внутри каждой темы располагают по алфавиту.

При решении задач ты встретишься с разными способами сортировки.

1

Расположи эти слова в словарном порядке: построй из всех этих слов такую последовательность, в которой слова стоят в словарном порядке.

зяблик ласточка воробей тетерев  
неясыть казарка дятел орёл  
какаду соловей иволга киви удоб  
журавль клёст горлица коршун  
буревестник малиновка аист петух  
фазан рябчик щегол



Обрати внимание, что все слова в этой задаче — это названия птиц.



2

Расположи слова в словарном порядке. При этом каждое слово пиши с новой строки и после каждого слова пропускай одну строку. Теперь напиши на пропущенных строках другие русские слова так, чтобы словарный порядок слов в последовательности (сверху вниз) сохранился.

олень	они	оперение	обвал	огарок
один	океан	овал	омлет	

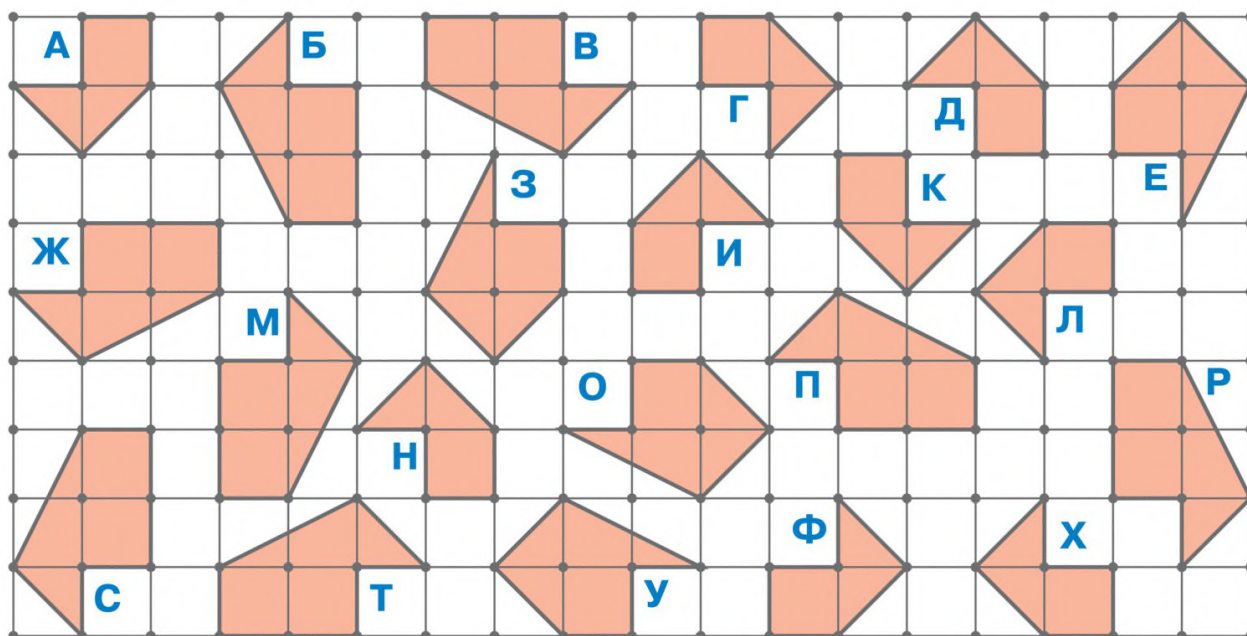
3

Расположи числа в порядке возрастания — построй из всех этих чисел такую последовательность, в которой каждое число (кроме первого) больше предыдущего.

12345	1234	98	89	123456
1234567	987	123	9876	12
98765	2345	23	234	987654

4

Классифицируй фигуры по одинаковости — выпиши имена многоугольников по группам так, чтобы в каждой группе все фигуры были одинаковыми, а две любые фигуры из разных групп были разными.



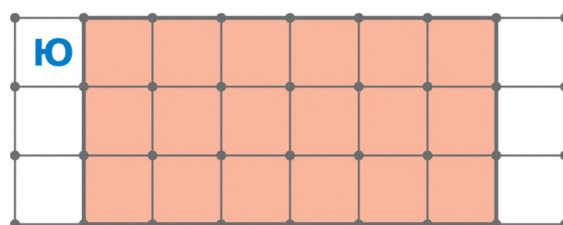
5

Напиши список учащихся своего класса:

- а) в словарном порядке фамилий (учащихся с одинаковыми фамилиями можно располагать в этом списке в любом порядке);
- б) в словарном порядке полных имён (учащихся с одинаковыми именами можно располагать в этом списке в любом порядке);
- в) упорядоченный по возрасту: начиная с самого старшего учащегося и заканчивая самым младшим (учащихся, родившихся в один день, можно располагать в этом списке в любом порядке).

6

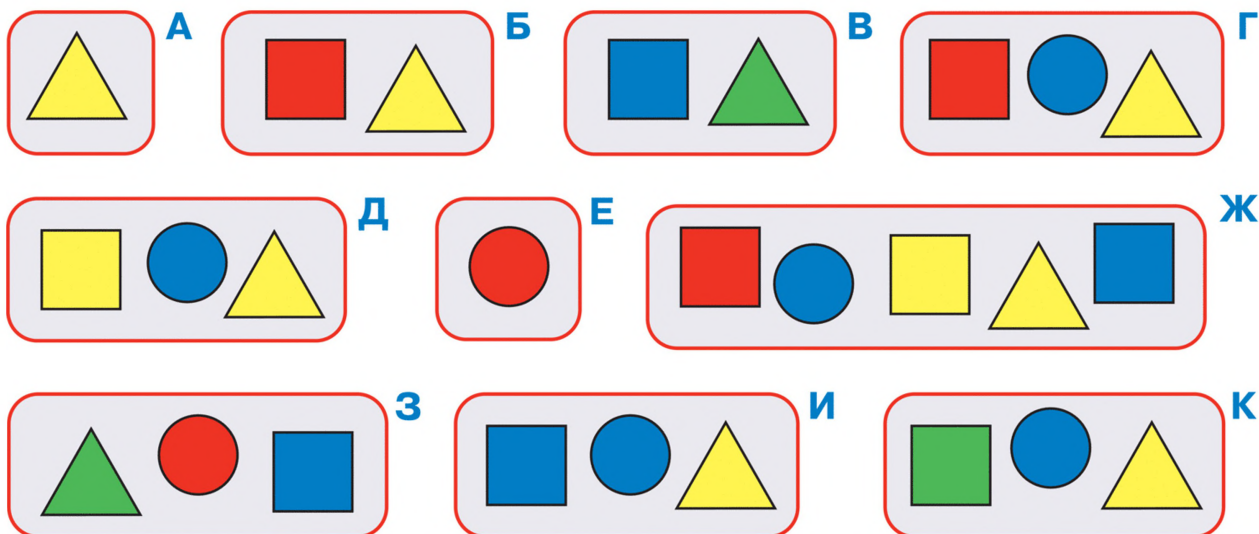
Нарисуй такой же прямоугольник по клеткам в тетради. Затем нарисуй, как разрезать прямоугольник Ю, чтобы получились четыре одинаковых многоугольника на сетке (с вершинами в узлах сетки). Найди площадь каждой части в единичных квадратах (ед. кв.).



7

Среди данных множеств найди:

- а) множество, равное объединению двух других данных множеств;
  - б) два каких-нибудь множества, пересечение которых равно множеству из одной бусины;
  - в) два каких-нибудь множества, пересечение которых равно множеству из двух бусин.
- Записывай полные ответы.





8

Слова в последовательности Ф упорядочены по некоторому правилу с использованием алфавитного порядка. Опиши это правило.

9

Классифицируй дорожные знаки — выпиши имена знаков по группам так, чтобы в одной группе были только предупреждающие знаки, в другой группе — только запрещающие, в третьей — только предписывающие, в четвёртой — все остальные, не вошедшие ни в какую из первых трёх групп. Чтобы уточнить, какой знак к какой группе относится, найди Правила дорожного движения в библиотеке или в Интернете.



Ф  
классики  
усики  
трусики  
ботики  
плечики  
бубенчики  
блинчики  
щипчики  
прыгалки  
посиделки  
опилки  
носилки  
чулки  
догонялки  
потёмки  
санки  
валенки  
ботинки  
гонки  
автогонки



10

Расположи названия городов Великобритании в словарном порядке.

Manchester	London	Cardiff
Liverpool	Edinburgh	Leeds
Birmingham	Plymouth	Glasgow
	Bristol	



В случае затруднений можешь воспользоваться английским алфавитом.

11

Расположи слова по следующему правилу (по этому правилу упорядочивают слова в так называемых обратных словарях):

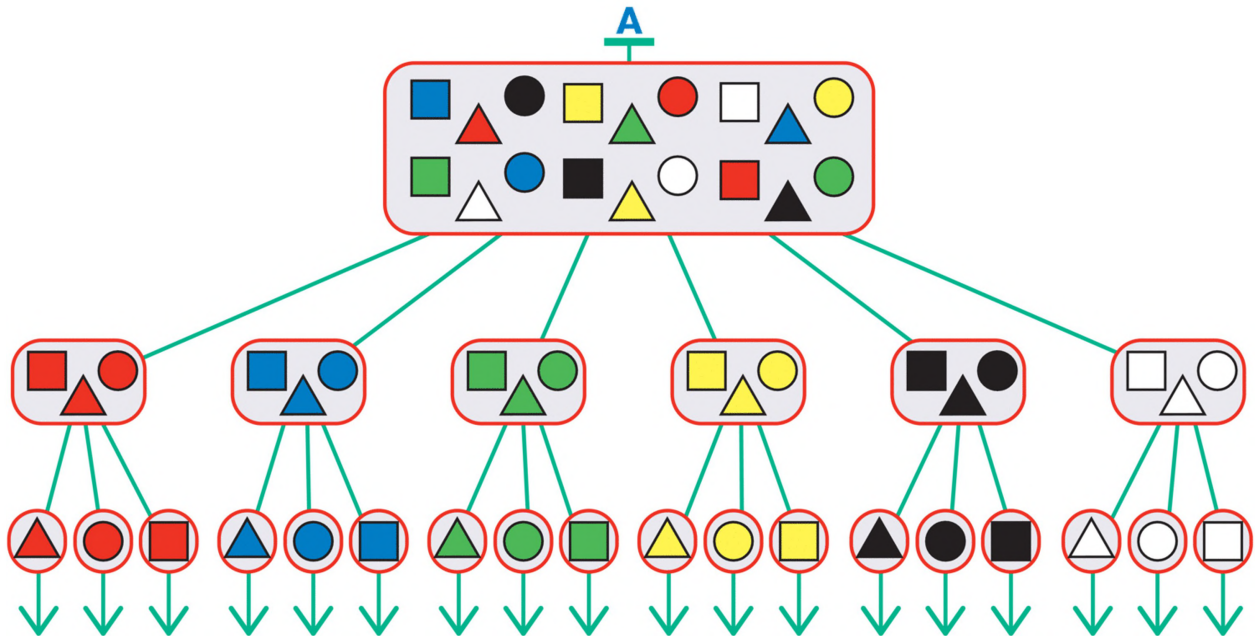
1. Сравним последние буквы двух слов: если эти буквы разные, то раньше будет идти то слово, последняя буква которого идёт раньше в алфавите.
2. Если у двух слов последние буквы одинаковые, то сравним предпоследние буквы: если эти буквы разные, то раньше будет идти то слово, предпоследняя буква которого идёт раньше в алфавите.
3. Если и предпоследние буквы у двух слов одинаковые, то сравним третьи с конца буквы: если эти буквы разные, то раньше будет идти то слово, третья с конца буква которого идёт раньше в алфавите.
4. Если и третьи с конца буквы у двух слов одинаковые, будем сравнивать четвёртые, пятые с конца и т. д., пока не дойдём до двух разных букв или пока одно из слов не закончится. В этом случае раньше идёт то слово, которое короче.

ударение	басок	дутый
квасок	плечо	
часок	куплет	
обезьянка	воздух	



# Дерево сортировки

Классифицируем множество всех бусин сначала по цветам, а затем в каждой полученной группе по форме. Такой процесс удобно изобразить в виде *дерева*:



Дерево А — **дерево сортировки** множества всех бусин. Вначале было множество всех бусин, состоящее из 18 элементов. В результате первого этапа классификации получилось 6 множеств, в каждом из этих множеств бусины одного цвета. В результате второго этапа классификации получились множества, состоящие из отдельных бусин.

Как видишь, дерево А позволило нам изобразить сразу весь процесс сортировки, при этом видны результаты каждого её этапа: результаты каждого этапа находятся на своём уровне дерева.

Вспомни: *начало* дерева мы обозначаем так же, как начало последовательности. Листья (элементы дерева, после которых нет следующих элементов) мы обозначаем так же, как последний член последовательности, — стрелкой.

В дереве после одного элемента может следовать *сразу несколько элементов*. Но каждый элемент дерева имеет *не больше одного предыдущего* элемента.

Мы называем в дереве следующий элемент *ребёнком*, а предыдущий — *родителем*. Элемент первого уровня дерева не может быть ребёнком ни для какого элемента, а лист дерева не может быть родителем ни для какого элемента.

Каждому листу дерева соответствует *последовательность* — последовательность элементов дерева, следующих друг за другом. Первый член такой последовательности — элемент первого уровня, а последний член — лист дерева. Последовательности, соответствующие разным листьям, могут оказаться одинаковыми.

Дерево, как мы его описали, позволяет удобно представить набор последовательностей элементов из бусин дерева. В математике употребляются и другие определения дерева.

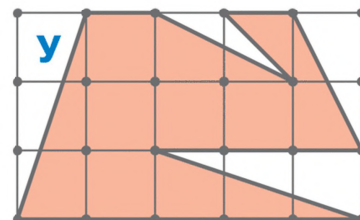
12

Построй в тетради дерево такой сортировки бусин, при которой все бусины группируются сначала по форме, а затем по цвету. Ответь на вопросы:

- а) Сколько в твоём дереве получилось элементов первого уровня?
- б) Сколько в твоём дереве получилось листьев?

13

Найди площадь многоугольника У.



14

Расположи слова в словарном порядке.

будильник      бубличек  
бугор      буженина      бубенец  
будни      бубен      бузина      будочник  
бублик      бубнить      будто      будочка  
будить      бубенчик      будущий      буйвол



15

Выпиши в порядке убывания последовательность всех двузначных чисел, которые делятся на 17.

16

Расположи слова в порядке обратного словаря (правило упорядочения приведено в задаче 11).

тренькать

баюкать

крякать

помелькать

убаюкать

булькать

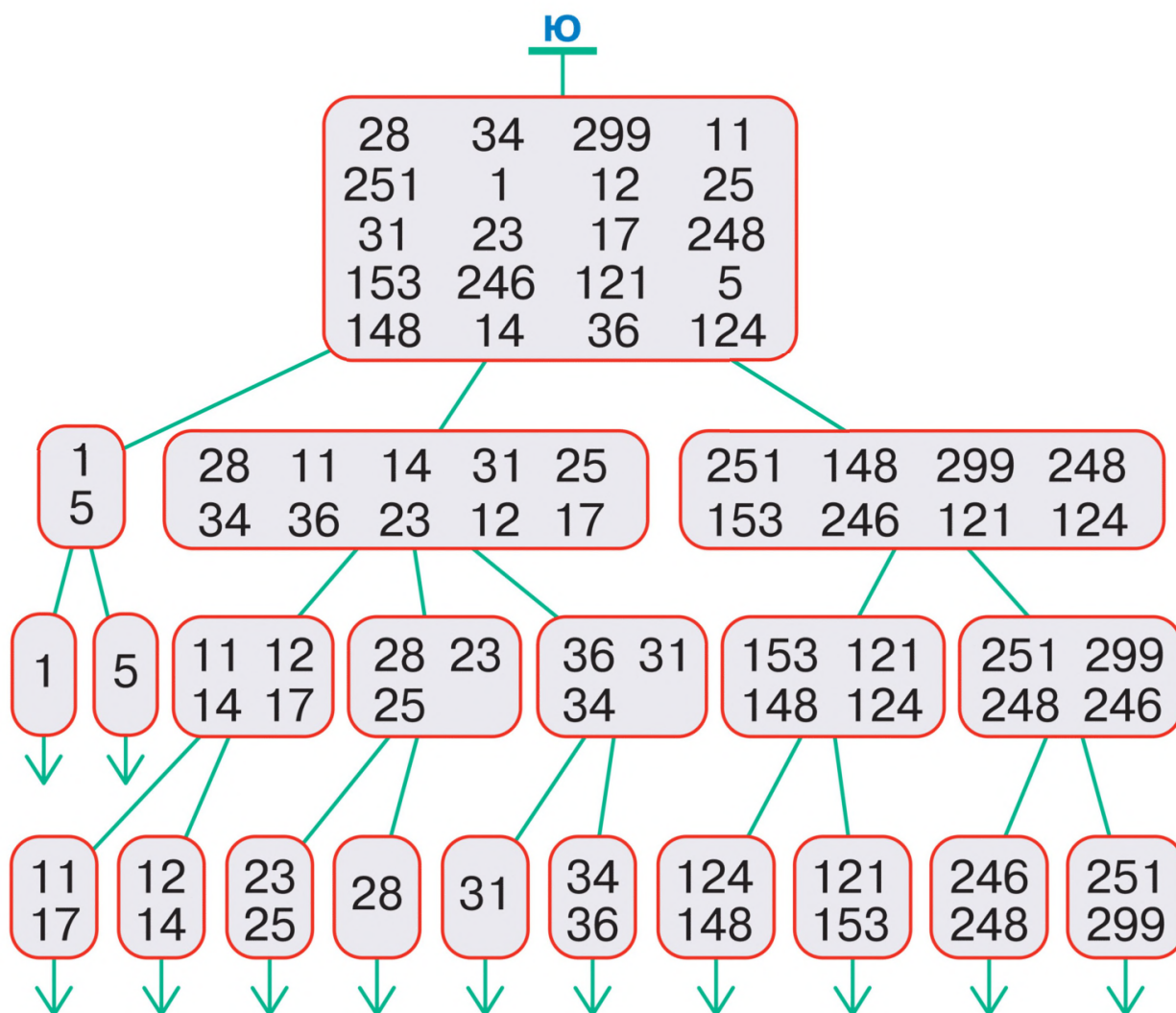
хрюкать

мелькать

дзинькать

17

Вот дерево Ю сортировки некоторого множества чисел. Сколько этапов было в этой сортировке? Для каждого этапа запиши правило, по которому сортировались числа.



В таблице даны русские слова в морфологическом представлении из лингвистического словаря (точки в слове делят слово на части, знак Ø означает отсутствие той или иной части слова). Выпиши из таблицы:

- а) все слова, классифицировав их по корням;
- б) все слова, классифицировав их по приставкам;
- в) все слова с корнем **раз/рез**;
- г) все слова с приставкой **раз/рас**;
- д) одно слово, которое будет единственным в группе, если рассортировать все слова таблицы сначала по корням, а потом по приставкам.

Слово	Приставка	Корень
об.лом.ов.к.а	об	лом/лам
на.рез.к.а	на	раз/рез
раз.рез.к.а	раз/рас	раз/рез
лом.к.а	Ø	лом/лам
за.кат.к.а	за	кат
об.раз. Ø	об	раз/рез
об.рез. Ø	об	раз/рез
об.рез.к.и	об	раз/рез
об.лом.к.и	об	лом/лам
кат.ок Ø	Ø	кат
об.лом. Ø	об	лом/лам
раз.лом. Ø	раз/рас	лом/лам
за.кат. Ø	за	кат
на.кат. Ø	на	кат
об.лам.ыв.а.т.ь	об	лом/лам
раз.рез.а.т.ь	раз/рас	раз/рез
рас.кат.а.т.ь	раз/рас	кат



19

Даны числа, классифицированные на четыре группы по некоторому правилу. Опиши это правило. Запиши ответ по образцу: «В группе А находятся числа, у которых...»

А

37	19
64	73 82

В

92	38	56
74	65	

С

95	77	86
68	59	

D

76	67
85	94 49

20

Реши задачу.

Каждый из трёх мальчиков написал 100 слов, после этого мальчики сравнили свои записи. Если слово встретилось хотя бы у двоих, то его вычёркивали из всех списков и вносили в новый список — «Список совпадающих слов». В результате



у первого мальчика в списке осталось 58 слов, у второго — 66, у третьего — 62. В «Списке совпадающих слов» оказалось 54 слова. Сколько было слов, которые встретились сразу в трёх списках?

21

Классифицируй слова по частям речи и внутри каждой группы расположи слова в словарном порядке. Рядом с каждой группой напиши, к какой части речи относятся слова этой группы.

выездной	квадратной	отказали	объехать
визг	указал	круг	выезжать
пробежка	квадрат	треугольному	отказ
пробежал	объезд	чудной	визжит
	круглая	выездка	указка

# Словари

*Словарь* — это справочная книга, которая содержит слова, расположенные в определённом порядке.

Уточним, что именно мы имеем в виду, когда говорим «слово словаря». Как ты знаешь, в информатике словом называют любую последовательность букв. Но наука о языке (она называется *языковедением* или *лингвистикой*) рассматривает не любые последовательности букв, а только те слова, которые есть в языке.

Кроме того, *слово* в словаре и *слово* в тексте — не одно и то же. В тексте следуют друг за другом формы слов (или, как говорят, *словоформы*). Разные формы слов могут относиться к одному словарному слову (например, *слон*, *слона*, *слонов* относятся к одному словарному слову *слон*) или к разным словам словаря (например, формы *слона*, *слонёнка*, *слонихой* относятся к разным словарным словам: *слон*, *слонёнок*, *слониха*).

Словари — это не просто сборники слов, а собрания каких-то специальных сведений о словах. Каждому слову в словаре отводится словарная статья, содержащая сведения об этом слове (в разных словарях разные, например, его перевод на другой язык или толкование, т. е. объяснение его смысла).

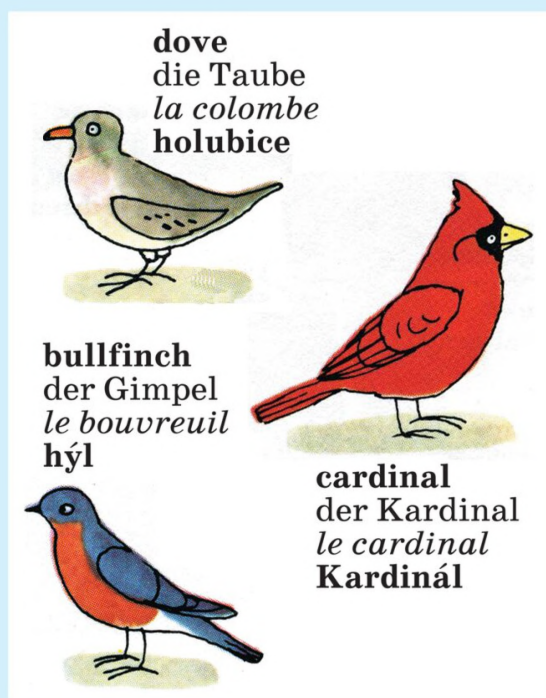
Ты пользуешься *двуязычным словарём*. В таком словаре основное содержание словарной статьи — это набор переводов слова на другой язык. Словарные статьи этого словаря упорядочены в словарном порядке алфавита первого языка: в немецко-русском словаре используется словарный порядок немецкого языка, а в русско-немецком — русского.

С некоторыми *одноязычными словарями*, например толковыми и орфографическими, ты тоже уже знаком. Бывают и другие одноязычные словари. Один из самых интересных одноязычных словарей — Грамматический словарь русского языка, составленный одним из крупнейших современных лингвистов А. А. Зализняком. В каждой его словарной статье содержится информация о грамматическом поведении слова — как оно склоняется, как спрягается, как у него изменяется при склонении или спряжении ударение и т. п. Вся эта информация записана сокращённо с помощью специальных словарных индексов. Слова в этом словаре стоят в *обратном* словарном порядке.



па́щенок	мо	з*а
воробье́нок	мо	з°а
соловьё́нок	мо	з°а
и́нок	мо	з°а
барви́нок	м	з*а
подсви́нок	мо	з*а
поеди́нок	м	з*а
обжи́нок	м	з*а
ожи́нок	м	з*а
блино́к	м	з*b
суглино́к	м	з*а
клино́к	м	з*b
закли́нок	м	з*а
подкли́нок	м	з*а
пино́к	м	з*b
мерино́к	мо	з*b
боти́нок	м	з*а <sup>②</sup>
полуботи́нок	м	з*а <sup>②</sup>
почи́нок	м	з*а
шино́к	м	з*b
челно́к	м	зb
звонóк	м	з*b
1-2 позвоно́к	м	з*b
поддо́нок	м	з*а

Фрагмент Грамматического словаря русского языка А. А. Зализняка



Фрагмент англо-немецко-французско-чешского детского словаря в картинках

Создаются также *многоязычные* переводные словари. Обычно это словари терминов по каким-то отдельным областям знаний: словарь математических терминов, словарь названий животных и растений и др.

Словари-энциклопедии (детские и взрослые) часто создаются в форме *словаря в картинках* и бывают и одноязычные, и многоязычные. Словарь в картинках может помочь, когда нужно найти, например, английское название какого-то предмета или животного, но вы забыли (а может быть, и не знали), как это называется по-русски.

Существуют и словари, в которых собраны все слова, встречающиеся в произведениях одного автора, например *Словарь языка Пушкина*. Есть *словари-энциклопедии*, например *Музыкальный энциклопедический словарь*, *Мифологический словарь*, *Исторический словарь*.

Почти все словари, которыми мы сегодня пользуемся, создавались вручную. Вручную создавался и *Грамматический словарь русского языка*, насчитывающий более 110 000 словарных статей. Вот что пишет его автор в предисловии к изданию: «Начало работы над ним относится к 1964 году. Нынешним молодым читателям



πολίτης, ου *м.р.* 1) гражданин;  
2) горожанин, житель города;  
соотечественник  
πολλάκις *нар.* много раз, много-  
кратно, часто  
πολλαπλασίονα *ср.р. мн.ч.* во мно-  
го раз больше  
πολυλογία, ας *ж.р.* многословие;  
длинная молитва  
πολυμερῶς *нар.* многосторонне,  
многообразно  
πολυποίκιλος, ου многосторон-  
ний, многообразный  
πολύς, πολλή, πολύ (*род.п.*  
πολλοῦ, ῆς, οῦ) 1. *прил.* 1) мно-  
гий, много, многочисленный;  
2) большой, значительный; 3) силь-  
ный, громкий (*о плаче*); богатый  
(*об урожае*); поздний (*о часе*);  
долгий, продолжительный (*о*

Фрагмент древнегреческо-русско-  
го словаря

**ИНФОРМАТИВНЫЙ**, -ая, -ое; -вен,  
-вна. Насыщенный информацией, хо-  
рошо информирующий. // *сущ. инфор-*  
**мативность**, -и, *ж.*

**ИНФОРМАТИКА**, -и, *ж.* Наука об общих  
свойствах и структуре научной инфор-  
мации, закономерностях ее создания,  
преобразования, накопления, передачи  
и использования.

**ИНФОРМАТОР**, -а, *м.* Тот, кто инфор-  
мирует. // *прил. информаторский*, -ая,  
-ое.

**ИНФОРМАЦИЯ**, -и, *ж.* 1. Сведения об  
окружающем мире и протекающих в нем  
процессах, воспринимаемые человеком  
или специальным устройством (*спец.*). *Тео-*  
*рия информации* (раздел кибернетики,  
изучающий способы измерения и переда-  
чи информации). 2. Сообщения, осведом-  
ляющие о положении дел, о состоянии че-  
го-н. *Научно-техническая и. Газетная и.*  
*Средства массовой информации* (печать,  
радио, телевидение, кино). ♦ **Генетиче-**  
**ская информация** (*спец.*) — совокуп-  
ность наследственных признаков, пере-  
даваемых от клетки к клетке, от орга-  
низма к организму. // *прил. информа-*  
**ционный**, -ая, -ое. *Информационное*  
*бюро. И. бюллетень*

Фрагмент Толкового словаря русского  
языка С. И. Ожегова

уже трудно представить себе, что эта работа делалась вручную. „Это же немыслимый абсурд — делать такую работу без компьютера“, — доводилось мне слышать. В действительности рабочим инструментом были четыре хлебных лотка, раздобытые в соседней булочной; в каждый входило по 25 тысяч карточек из тонкой бумаги». И все эти тысячи карточек сортировались вручную, притом что хороший темп опытного сортировщика составлял тысячу карточек в час. Сегодня такую сортировку поручают компьютерной программе.

Умение пользоваться словарём является важной частью информационной культуры, о которой мы говорили в самом начале этого учебника. Это относится не только к умению быстро найти нужное слово в словаре, но и к умению правильно выбрать словарь, а также к умению правильно истолковать и использовать информацию, взятую из словаря.



22

Вот русско-английский словарик названий рыб. Составь англо-русский словарик этих же названий. Не забудь упорядочить слова в словарном порядке.

*Акула* — shark

*Камбала* — plaice

*Морской конёк* — seahorse

*Окунь* — bass

*Рыба-меч* — swordfish

*Сёмга* — salmon

*Сом* — sheat

*Треска* — cod

*Тунец* — tuna

*Угорь* — eel

*Форель* — trout

*Щука* — pike

23

Когда составляли некоторый толковый словарь русского языка, завели два набора карточек. На одних карточках писали глаголы, а на других — их значения. Случилось так, что карточки перепутали. Восстанови соответствие глаголов и их значений.

УДАЛЯТЬСЯ

ЕХАТЬ

ДОГОНЯТЬ

ИДТИ

ОТСТАВАТЬ

Лицо перемещается из одного пункта в другой, находясь на движущемся транспортном средстве, и это перемещение входит в число целей перемещающегося.

Объект перемещается, и расстояние между ним и другим объектом увеличивается.

Два объекта перемещаются в одном и том же направлении, расстояние между ними увеличивается, и один из них находится позади другого.

Живое существо перемещается из одного пункта в другой, переступая ногами и ни в какой момент не утрачивая полностью контакт с поверхностью, по которой оно перемещается.

Два объекта перемещаются в одном и том же направлении, расстояние между ними сокращается, и один из них находится позади другого.

24

Расположи слова в порядке обратного словаря (правило упорядочения приведено в задаче 11).

понемножку	смолоду	понемногу	сбоку
взаправду	сразу	между	наяву
неподалёку	повсюду	внизу	наружу

25

Вот латинско-русский словарьк названий птиц. Составь русско-латинский словарьк этих названий. Не забудь упорядочить слова в словарном порядке.

*Bombycilla garrulus* — свиристель

*Corvus frugileus* — грач

*Emberiza aureola* — дубровник

*Erithacus rubecula* — малиновка

*Ficedula hypoleuca* — мухоловка-пеструшка

*Fringilla coelebs* — зяблик

*Loxia curvirostra* — клёст-еловик

*Nucifraga caryocatactes* — кедровка

*Phoenicurus phoenicurus* — горихвостка

*Phylloscopus inornatus* — пеночка-зарничка

*Riparia riparia* — береговушка

*Spinus spinus* — чиж

26

Построй последовательность чисел по инструкции.

1. Запиши первый член последовательности: однозначное число, большее 1.
2. Запиши второй член последовательности: однозначное число, большее 2, не равное первому члену последовательности.
3. Каждый следующий член находи по правилу: искомое число равно сумме последнего и предпоследнего из уже построенных членов последовательности.
4. Строй последовательность до тех пор, пока в последовательности не появится трёхзначное число.



27

Выпиши названия всех месяцев года на любых двух иностранных языках.



Выписанные переводы и русские названия разместить в таблице: в каждом столбце должны быть названия месяцев на одном языке, а в каждой строке — название одного месяца на разных языках. Для работы тебе понадобятся двуязычные словари.

28

Классифицируй названия государств по их принадлежности к Европе, Африке или Южной Америке, а затем в каждой группе расположи слова в словарном порядке.

Россия	Финляндия	Венесуэла
Чехия	Боливия	Колумбия
Франция	Молдова	Гвинея
Беларусь	Ватикан	Эстония
Танзания	Бразилия	Ангола
Польша	Парагвай	Латвия
Алжир	Аргентина	Чили

29

Реши задачу.

По соседству стоят два города: город Рыцарей и город Лжецов, жители которых ездят друг к другу в гости. При этом рыцари всегда говорят правду, а лжецы всегда лгут. В одном из этих городов встретились двое, и между ними состоялся такой разговор:

Первый. Жители этого города — рыцари. А ты — лжец.

Второй. Нет, я рыцарь. Каждый мой сосед скажет, что я рыцарь.

Кто был Первый и кто Второй? В каком городе состоялся этот разговор?



30

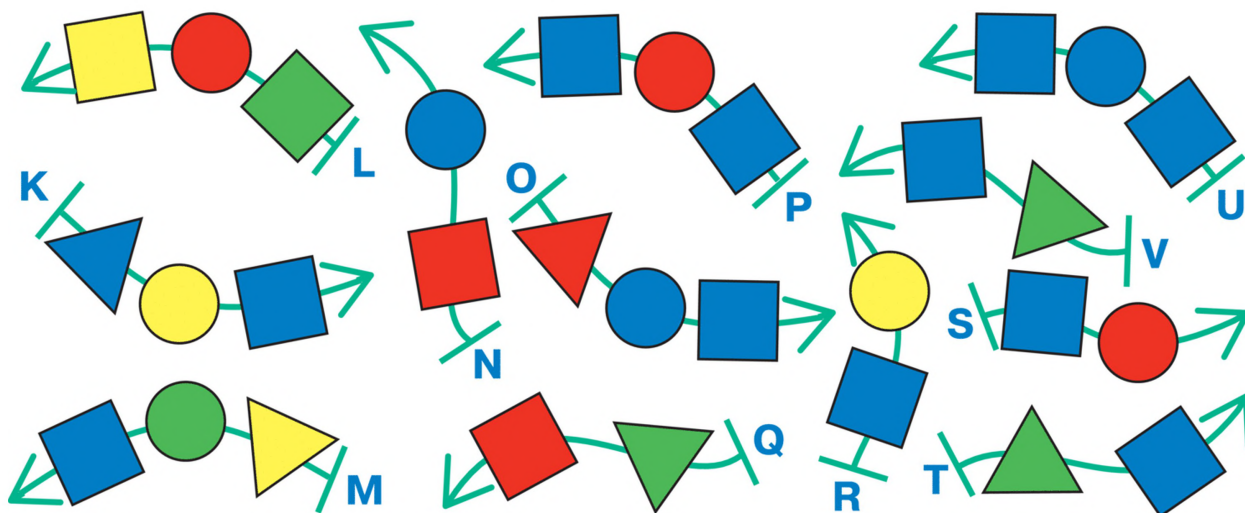
Вот таблица с приближённым описанием свойств планет Солнечной системы. Выпиши планеты:

- а) в порядке возрастания диаметра;
- б) в порядке убывания расстояния от Солнца;
- в) в порядке убывания продолжительности года (периода обращения планеты вокруг Солнца).

Планета	Диаметр	Среднее расстояние от Солнца	Продолжительность года
Меркурий	4800 км	58 млн км	88 суток
Венера	12 000 км	108 млн км	225 суток
Земля	12 800 км	150 млн км	365 суток
Марс	6700 км	228 млн км	687 суток
Юпитер	140 000 км	778 млн км	12 лет
Сатурн	120 000 км	1431 млн км	29 лет
Уран	51 000 км	3 млрд км	84 года
Нептун	49 500 км	4,5 млрд км	165 лет
Плутон	2300 км	6 млрд км	248 лет

31

Придумай классификацию последовательностей по четырём группам. Выпиши имена последовательностей этих групп. Опиши признак, по которому собрана каждая группа.





# Исполнители и алгоритмы

Мы называли **исполнителем** любое устройство, которое способно выполнять определённый набор действий. Все команды, которые понимает исполнитель, образуют **систему команд** этого исполнителя. В каждый момент исполнитель

находится в одном из возможных для него **состояний**. **Команды-приказы** изменяют состояние исполнителя. При выполнении **команды-запроса** исполнитель сообщает информацию о своём состоянии, состояние исполнителя не изменяется. **Начальным состоянием** исполнителя мы называем его состояние перед выполнением программы.

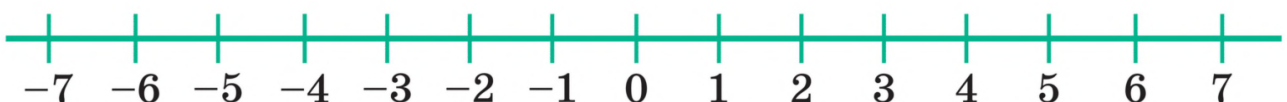
Ты знаком с исполнителями *Водолей*, *Удвоитель*, *Кузнечик* и *Робот*.

В системе команд *Водолея* команд-запросов нет. Состояние *Водолея* (при заданной вместимости сосудов) описывается тем, сколько мер воды в данный момент налито в каждый сосуд.

В системе команд *Удвоителя* команд-запросов нет. Состояние исполнителя *Удвоитель* описывается только одним числом — это число отображается на экране. В начальном состоянии *Удвоителя* на экране обычно отображается число 0, но в задачах бывают

и другие начальные состояния.

Исполнитель *Кузнечик* работает на числовой прямой:



Эта числовая прямая получилась объединением двух числовых лучей: один числовой луч построен от точки «0» вправо,

## Система команд исполнителя *Водолей*

наполни А  
наполни В  
наполни С  
перелей из А в В  
перелей из В в С  
перелей из А в С  
перелей из В в А  
перелей из С в В  
перелей из С в А  
вылей А  
вылей В  
вылей С

## Система команд исполнителя *Удвоитель*

умножь на 2  
прибавь 1

а другой числовой луч построен от точки «0» влево. На левом числовом луче числа помечены знаком «-». Скоро ты узнаешь, что такие числа называются отрицательными.

*Кузнечик* — это не один исполнитель, а много похожих исполнителей. *Кузнечики* различаются системами команд. У каждого *Кузнечика* в системе ровно две команды (вперёд ... и назад ...), но числа в командах (количество шагов, на которые прыгает *Кузнечик*) могут быть разными. Пример системы команд исполнителя *Кузнечик* приведён справа.

**Пример  
системы команд  
исполнителя  
*Кузнечик***

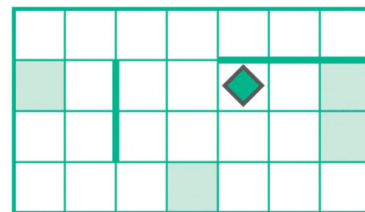
вперёд 3  
назад 2

*Кузнечик* в начальном состоянии обычно стоит в точке «0», но в некоторых задачах может быть задано и другое начальное состояние.

Исполнитель *Робот* работает на прямоугольном поле, разбитом на клетки. Между некоторыми клетками поля могут стоять стены (на рисунке стены отмечены жирными линиями). Некоторые клетки поля могут быть закрашены. Сам *Робот* всегда занимает ровно одну клетку поля.

Исполнитель *Робот* умеет выполнять 17 команд: 5 команд-приказов и 12 команд-запросов. Мы изучили пока только команды-приказы: вверх, вниз, вправо, влево, закрасить.

По командам вверх, вниз, вправо, влево *Робот* перемещается в соседнюю клетку поля в указанном направлении. Если на пути *Робота* оказывается стена, команда не может быть выполнена: происходит отказ. В этом случае *Робот* команду не выполняет и выдаёт сообщение об ошибке. Например, из состояния, показанного на рисунке, *Робот* не может выполнить команду вверх.



По команде закрасить *Робот* закрашивает ту клетку, в которой стоит. Если эта клетка уже была закрашена, она останется закрашенной, т. е. команда будет выполнена, но никаких видимых изменений при этом не произойдёт.

В быту человек чаще всего управляет устройствами-исполнителями **непосредственно**: даёт исполнителю команду, исполнитель выполняет её, затем человек смотрит на результат и даёт следующую команду и т. д. При этом человек выбирает следующую команду в зависимости от результата выполнения

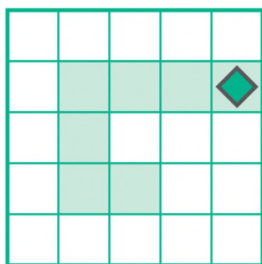


предыдущей команды и состояния исполнителя после её выполнения.

Однако часто возникают ситуации, когда непосредственное управление неудобно или даже невозможно. В таких случаях можно составить план действий (иногда говорят «программу действий»), в котором будет указано, какие команды нужно давать исполнителю. Такой план называется **алгоритмом**. Алгоритм должен быть описан так, чтобы его мог выполнить компьютер. Для записи алгоритмов есть специальные языки, они называются **языками программирования**. Алгоритм, записанный на языке программирования, называют **программой**.

Выполняя программу, компьютер читает её и даёт команды исполнителю. Какие выдавать команды и когда, компьютер узнаёт из программы. В программе могут быть **команды-запросы**. В этом случае команда, которую компьютер даст исполнителю, будет зависеть от ответа исполнителя на запрос.

В учебнике для записи алгоритмов мы используем **школьный алгоритмический язык** и для краткости называем его **алгоритмический язык**. С этим языком ты познакомился в 5 классе. Например, вот алгоритм крючок и поле *Робота* после выполнения этого алгоритма:



**алг** крючок

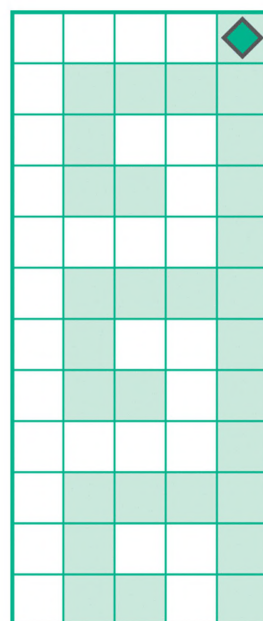
<b>дано</b>		слева от Робота
		не меньше 3
		клеток поля,
		снизу не меньше
		2 клеток
<b>надо</b>		Робот нарисовал
		крючок и
		вернулся назад

**нач**

закрасить  
влево  
закрасить  
влево  
закрасить  
влево  
закрасить  
вниз  
закрасить  
вниз  
закрасить  
вправо  
закрасить  
вправо  
вверх  
вверх

**кон**

KOH





Наряду с обычными командами, в алгоритме *узор* есть команда *крючок* — это команда вызова алгоритма *крючок*. Прочитав эту команду, компьютер последовательно отдаёт *Роботу* все команды, предписанные алгоритмом *крючок*. После этого компьютер переходит к следующей строке алгоритма *узор*. В этом случае говорят, что алгоритм *крючок* является **вспомогательным алгоритмом** для алгоритма *узор*.

Любой алгоритм можно использовать как вспомогательный. Для этого достаточно указать его имя в качестве команды в каком-то другом алгоритме. Использование уже написанных алгоритмов как вспомогательных позволяет свести новую задачу к уже решённым.

Справа приведены два алгоритма для *Удвоителя*: *вычисления1* и *вычисления2*. При этом алгоритм *вычисления1* используется как вспомогательный в алгоритме *вычисления2*.

Что нам нужно знать об алгоритме, чтобы мы могли его использовать как вспомогательный?

1. Необходимо знать **имя** этого алгоритма, чтобы можно было записать это имя в качестве команды.

2. Необходимо знать, из какого состояния исполнителя этот алгоритм может быть исполнен.

3. Необходимо знать, что получится в результате выполнения вспомогательного алгоритма — как изменится состояние исполнителя.

Вся эта информация должна содержаться в заголовке алгоритма — в строках **алг**, **дано**, **надо**. При этом для использования алгоритма как вспомогательного необязательно знать, какую именно последовательность команд он содержит.

Таким образом, заголовок алгоритма содержит всю информацию, необходимую для использования этого алгоритма как вспомо-

```
алг вычисления1
  дано |
  надо |
нач
  умножь на 2
  умножь на 2
  умножь на 2
  умножь на 2
кон
```

```
алг вычисления2
  дано |
  надо |
нач
  вычисления1
  прибавь 1
  умножь на 2
  прибавь 1
  умножь на 2
  прибавь 1
  умножь на 2
кон
```

гательного. Поэтому при решении задач старайся писать комментарии в строках **дано** (начальное состояние, если оно задано, или ограничения, если они известны) и **надо** (то, что нужно получить в результате выполнения алгоритма, если это указано в задаче).

32

Какое число будет отображаться на экране *Удвоителя* после выполнения им алгоритма *вычисления<sub>2</sub>* (с. 26) из начального состояния «1»? Измени одну команду во вспомогательном алгоритме *вычисления<sub>1</sub>* так, чтобы в результате выполнения алгоритма *вычисления<sub>2</sub>* из начального состояния «1» на экране *Удвоителя* отобразилось число «94». Напиши исправленный вариант алгоритма *вычисления<sub>1</sub>*.

33

Составь алгоритм для *Кузнечика* с системой команд *вперёд 7*, *назад 5*, который переводит *Кузнечика* на один шаг *вперёд* на числовой прямой — например из точки «4» переводит *Кузнечика* в точку «5». Дай алгоритму имя *шаг вперёд*.  
Теперь, используя алгоритм *шаг вперёд* как вспомогательный, составь алгоритм, в котором не больше 3 команд и который переводит *Кузнечика* на 8 шагов *вперёд*.

34

Используя алгоритм *шаг вперёд*, составленный при решении задачи 33, как вспомогательный, составь алгоритм, в котором ровно 3 команды и который переводит *Кузнечика*: а) на 3 шага *вперёд*; б) на 9 шагов *вперёд*; в) на 15 шагов *вперёд*.

35

Составь алгоритм для *Водолея* с вместимостью сосудов 3, 5 и 11 мер, после выполнения которого в сосуде А оказывается ровно 1 мера воды, а сосуды В и С будут пустыми. Дай алгоритму имя *налей в А 1 меру*.  
Теперь, используя алгоритм *налей в А 1 меру* как вспомогательный, составь алгоритм, в котором не больше 4 команд и в результате выполнения которого сосуды А и В оказываются пустыми, а в сосуде С налито ровно 2 меры воды.  
Для проверки работы своих алгоритмов заполняй такую таблицу:

Команда	А (3 меры)	В (5 мер)	С (11 мер)



36

Используя алгоритм налей в А 1 меру (задача 35) как вспомогательный, составь:

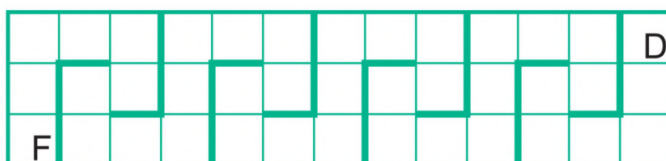
а) алгоритм, в котором ровно 4 команды и в результате выполнения которого сосуды А и С оказываются пустыми, а в сосуде В налито ровно 4 меры воды;

б) алгоритм, в котором ровно 4 команды и в результате выполнения которого сосуды А и В оказываются пустыми, а в сосуде С налито ровно 9 мер воды.

Для проверки работы своих алгоритмов заполняй таблицу, (см задачу 35).

37

Составь алгоритм, в котором не больше 6 команд, который переводит *Робота* из точки F в точку D. Для этого сначала построй вспомогательный алгоритм.



38

Даны алгоритмы фрагмент и фигура (справа). Напиши комментарий для **дано** каждого из двух алгоритмов: укажи, каким должно быть поле и в какой клетке поля должен находиться *Робот* в начальном положении, чтобы при выполнении алгоритма не произошло отказа.

Нарисуй результат выполнения алгоритма *фигура*, считая, что весь лист тетради — это поле *Робота*. Выбери начальное положение *Робота*, 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 алгоритма.

алг фрагмент

дано |

надо |

нач

закрасить

вправо

закрасить

вправо

закрасить

вправо

закрасить

вниз

закрасить

вниз

закрасить

вниз

закрасить

влево

закрасить

влево

закрасить

влево

закрасить

вправо

вправо

вправо

вверх

кон

алг фигура

дано |

надо |

нач

фрагмент

фрагмент

фрагмент

фрагмент

фрагмент

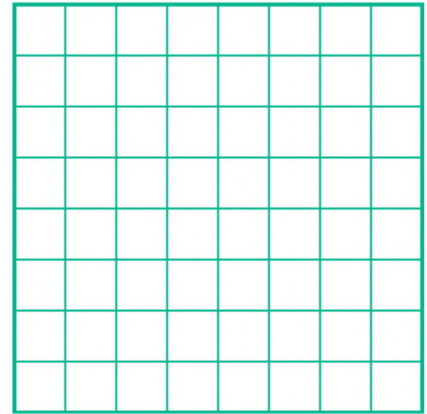
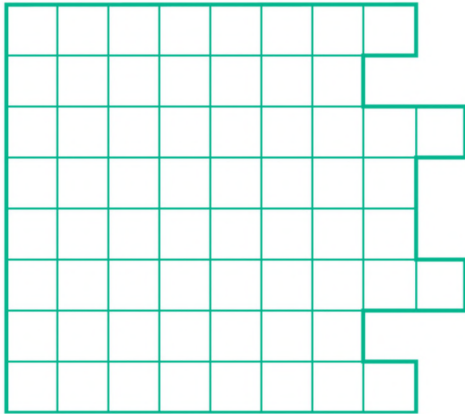
кон

39

Сколько команд закрасить компьютер даст *Роботу* при выполнении алгоритма фигура из задачи 38? Сколько клеток при этом будет закрашено дважды?

40

Подумай, как разрезать фигуру на две части так, чтобы из этих частей можно было сложить квадрат  $8 \times 8$ . Нарисуй линии разреза на фигуре слева и границы между частями на квадрате справа.



41

Построй дерево сортировки множества чисел  $A$ , в которой на первом этапе числа классифицируются по принадлежности к сотням (числа первой сотни, числа второй сотни и т. д.), на втором этапе числа классифицируются по принадлежности к десяткам (числа первого десятка сотни, числа второго десятка сотни и т. д.), а на третьем этапе — по чётности и нечётности.

A

237	600	100
91	200	60
541	392	297
50	595	53
549	117	199
120	193	240
300	45	550

42

Вот русско-английский словарь названий фруктов. Составь англо-русский словарь этих названий. Не забудь упорядочить слова в словарном порядке.

*Айва* — quince

*Ананас* — pineapple

*Банан* — banana

*Гранат* — pomegranate

*Грейпфрут* — grapefruit

*Груша* — pear

*Инжир* — fig

*Киви* — kiwi fruit

*Лимон* — lemon

*Мандарин* — mandarin

*Хурма* — persimmon

*Яблоко* — apple



43

Рассмотри эмблемы различных ведомств России. Найди две одинаковые фигурки.





# Дерево перебора вариантов.

## Дерево перебора подмножеств

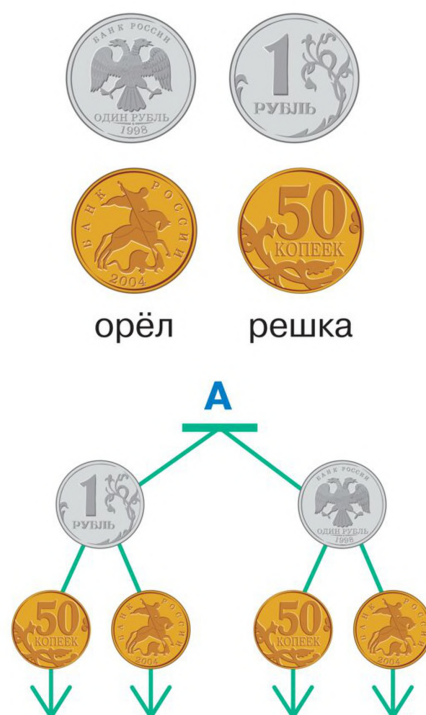
Мы познакомились с деревьями сортировки. Ещё один вид деревьев, которые часто используются в информатике, — *деревья перебора вариантов*. Чтобы построить множество всех возможных вариантов какого-либо процесса, не потеряв ни одного из вариантов и не добавив лишних, очень полезно построить дерево.

Задача 1. Если сначала подкинуть одну монету, а потом другую — какими способами они могут упасть на стол?

Монета может упасть на стол двумя способами — орлом или решкой.

Чтобы выписать все возможные варианты, построим **дерево перебора вариантов А**. На первом уровне дерева поместим все способы падения первой монеты — орёл и решка. На втором уровне после каждого элемента первого уровня поместим все способы падения второй монеты.

В дереве А всего 4 последовательности, и все они разные. В каждой последовательности из дерева А первый член — это одна из сторон первой монеты, а второй член — одна из сторон второй монеты.

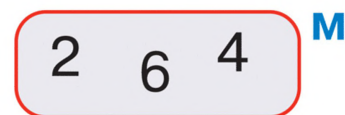


Множество всех последовательностей из дерева А и есть множество всех способов, которыми могут упасть на стол две монеты, — таких способов 4:

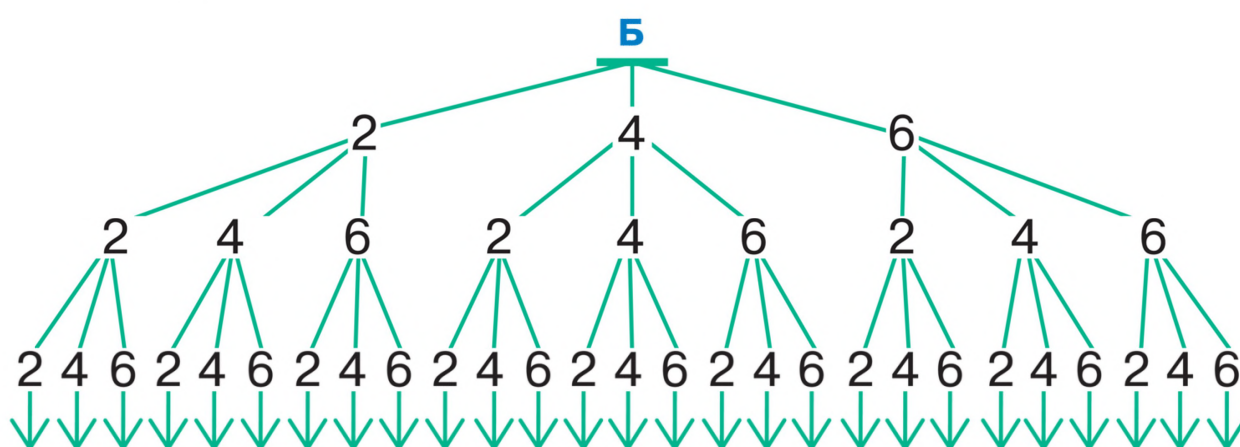




Задача 2. Найти все трёхзначные числа, в записи которых участвуют только цифры из множества М (возможно, с повторениями).



Для решения этой задачи построим дерево Б, множество всех последовательностей из которого будет её решением. Три уровня дерева Б будут соответствовать трём разрядам наших трёхзначных чисел. В искомым числах в каждом разряде может стоять любая цифра из множества М. Поэтому на первом уровне помещаем все цифры из множества М, на втором и на третьем уровнях помещаем цифры так, чтобы множество всех детей каждого родителя было равно множеству М.

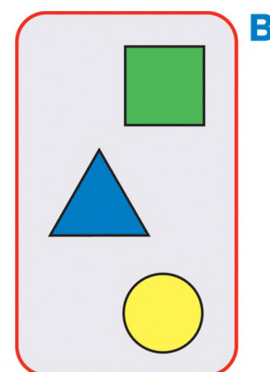


Множество последовательностей дерева Б и является решением задачи.

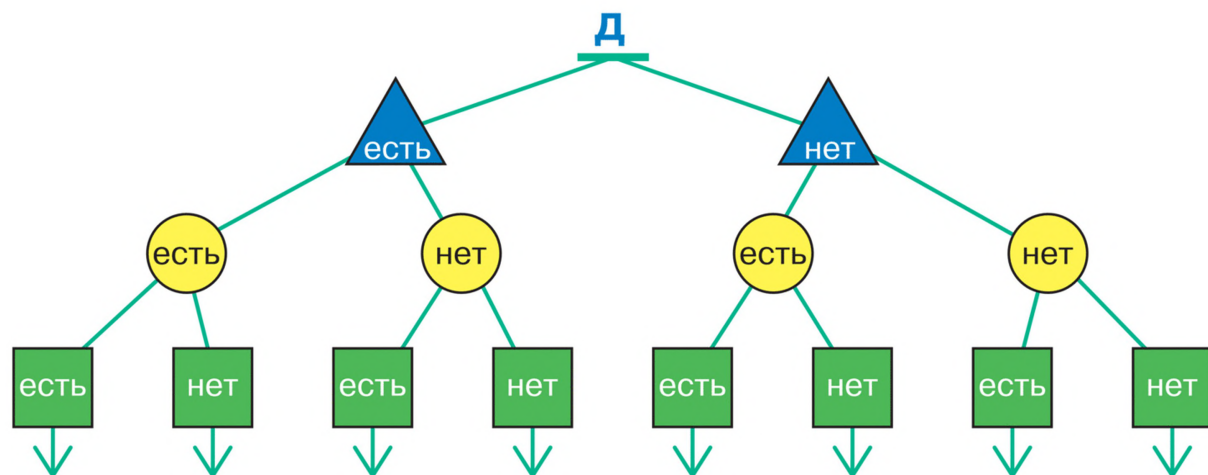
Построение дерева поможет нам и в построении **всех подмножеств данного множества**.

Задача 3. Построить все подмножества множества В.

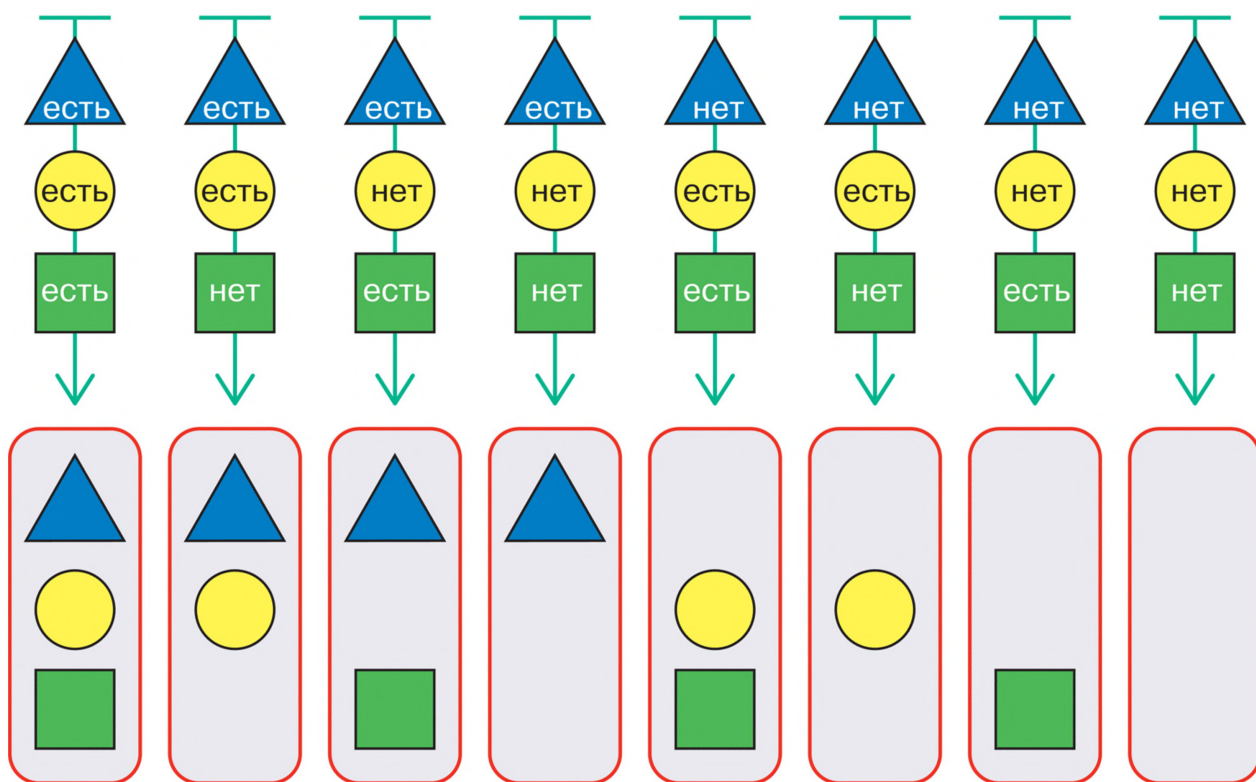
Для каждого из трёх элементов множества В все подмножества можно разделить на две группы: те, которые содержат данный элемент, и те, которые его не содержат. Используя это, будем строить дерево Д. Сначала выберем какой-нибудь один элемент множества В, например синюю треугольную бусину. На первом уровне дерева Д поместим знаки «есть синяя треугольная бусина» и «нет синей треугольной бусины». После каждого элемента первого уровня поместим



знаки «есть жёлтая круглая бусина» и «нет жёлтой круглой бусины». После каждого элемента второго уровня поместим знаки «есть зелёная квадратная бусина» и «нет зелёной квадратной бусины». Получим дерево Д.



Члены каждой последовательности из дерева Д изображают одно из подмножеств множества В. При этом для каждого возможного подмножества в дереве Д есть соответствующая ему последовательность.





44

Сначала подкинули одну монету, затем вторую, потом третью (все монеты разные). Сколькими способами могут упасть на стол эти три монеты? Построй дерево перебора вариантов. Сколько всего существует вариантов падения трёх разных монет?



Конечно, рисовать монеты не надо — можно просто рисовать круги с буквой О (для орла) и Р (для решки). Круги для разных монет рисуй разными цветами.

45

Сколько всего существует чётных двузначных чисел, в записи которых нет цифр 0, 2 и 3? Построй дерево перебора вариантов. Пользуясь деревом, выпиши все такие числа.

46

Построй все подмножества множества  $W$ .

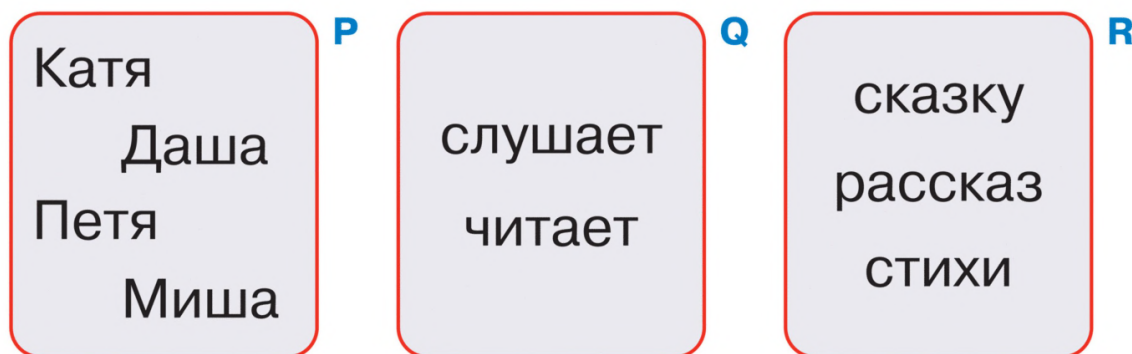


Для этого сначала построй дерево перебора подмножеств.



47

Для трёх множеств слов построй множество всех последовательностей слов длины 3, таких, что первый член последовательности — слово из множества  $P$ , второй — слово из множества  $Q$ , третий — слово из множества  $R$ . Для решения построй дерево перебора вариантов.



48

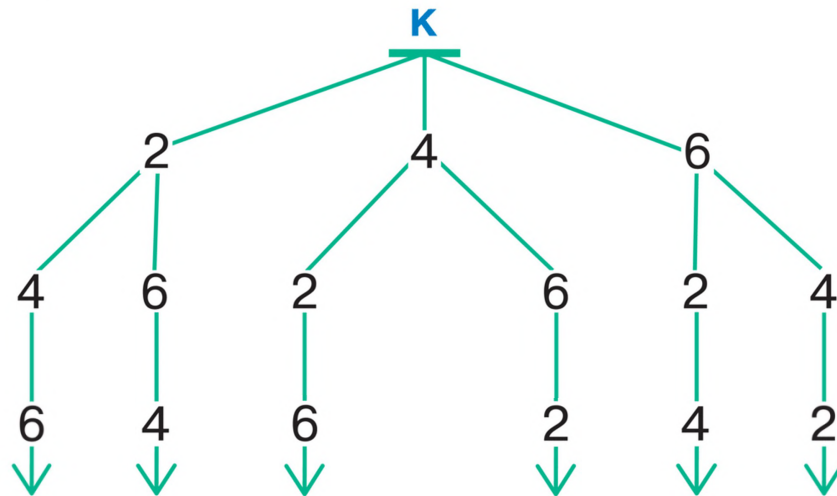
Реши задачу, построив дерево перебора вариантов.

У Вовы были в кармане 4 монеты: 1 к., 5 к., 10 к., 50 к. Он достал из кармана некоторое количество монет (он мог достать и все монеты, мог не достать и ни одной). Сколько денег оказалось в руке у Вовы?

Укажи все возможные варианты.

49

Сформулируй задачу, решению которой поможет дерево перебора вариантов К.



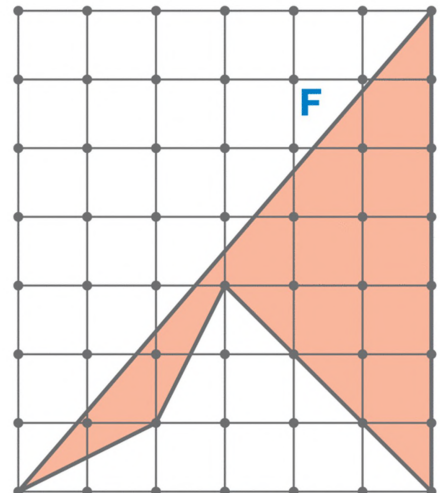
50

а) Построй алгоритм для *Удвоителя*, в котором меньше 8 команд. После выполнения этого алгоритма на экране должно отобразиться число 15.

б) Построй алгоритм для *Удвоителя*, в котором меньше 12 команд. После выполнения этого алгоритма на экране должно отобразиться число 1024.

51

Найди площадь многоугольника F.

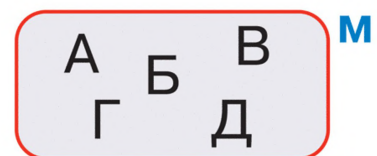


52

Сколько всего существует нечётных трёхзначных чисел, больших чем 500 и таких, что сумма их цифр равна 9? Построй дерево перебора вариантов. Пользуясь деревом, выпиши все такие числа.

53

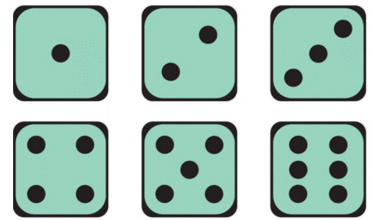
Построй множество всех последовательностей из двух разных букв, в записи которых участвуют только буквы из множества М. Сначала построь дерево перебора вариантов.





54

Сначала подкинули монету, а затем бросили игральную кость. Сколькими способами могут упасть монета и игральная кость? Построй дерево перебора вариантов.



Игральная кость может упасть на стол шестью способами — сверху одной из шести своих граней (см. рисунок).

55

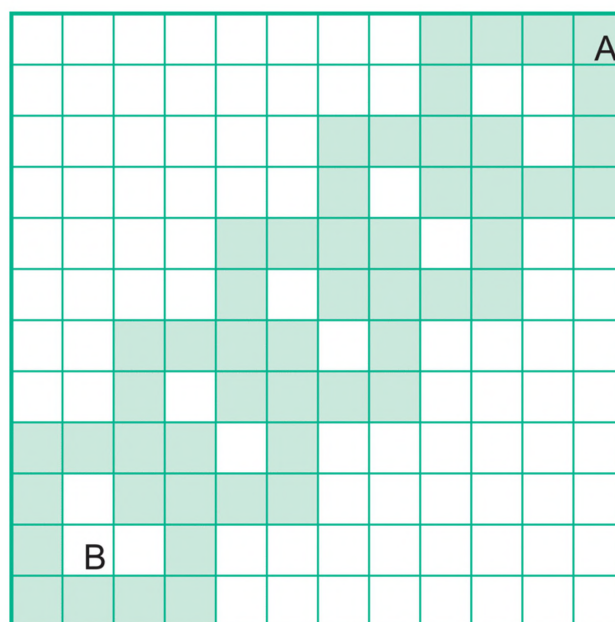
Построй все возможные последовательности длины 3, составленные только из нулей и единиц.

56

Сколько разных чисел можно получить, переставляя цифры числа 543? Построй дерево перебора вариантов.

57

Измени вспомогательный алгоритм *фрагмент* из задачи 38 так, чтобы при выполнении алгоритма *фигура Робот* переместился из А в В и закрасил клетки, как на рисунке ниже. Напиши исправленный вариант алгоритма *фрагмент*.  
Сколько команд закрасить компьютер даст *Робота* при выполнении алгоритма *фигура* с исправленным вспомогательным алгоритмом *фрагмент*? Сколько клеток при этом будет закрашено дважды?



58

Рассмотри меню, которое предлагается посетителям кафе «Таверна». Сколько вариантов завтраков (включающих яичницу или кашу и один напиток) могут заказать посетители кафе «Таверна»? Построй дерево перебора вариантов.

Сколько вариантов обедов (включающих один салат, суп, одно второе блюдо, один десерт и один напиток) могут заказать посетители кафе «Таверна»? Построй дерево перебора вариантов.

## МЕНЮ

### Завтрак



*Яичница*



*Каша овсяная*

### Салаты



*Салат «Цезарь»*



*Салат мясной*

### Напитки



*Молоко  
(только по утрам)*



*Чай*



*Кофе*

### Супы



*Супчик дня*

### Вторые блюда



*Курица  
по-корсикански*



*Биточки*



*Шашлык*



*Судак  
по-польски*

### Десерт

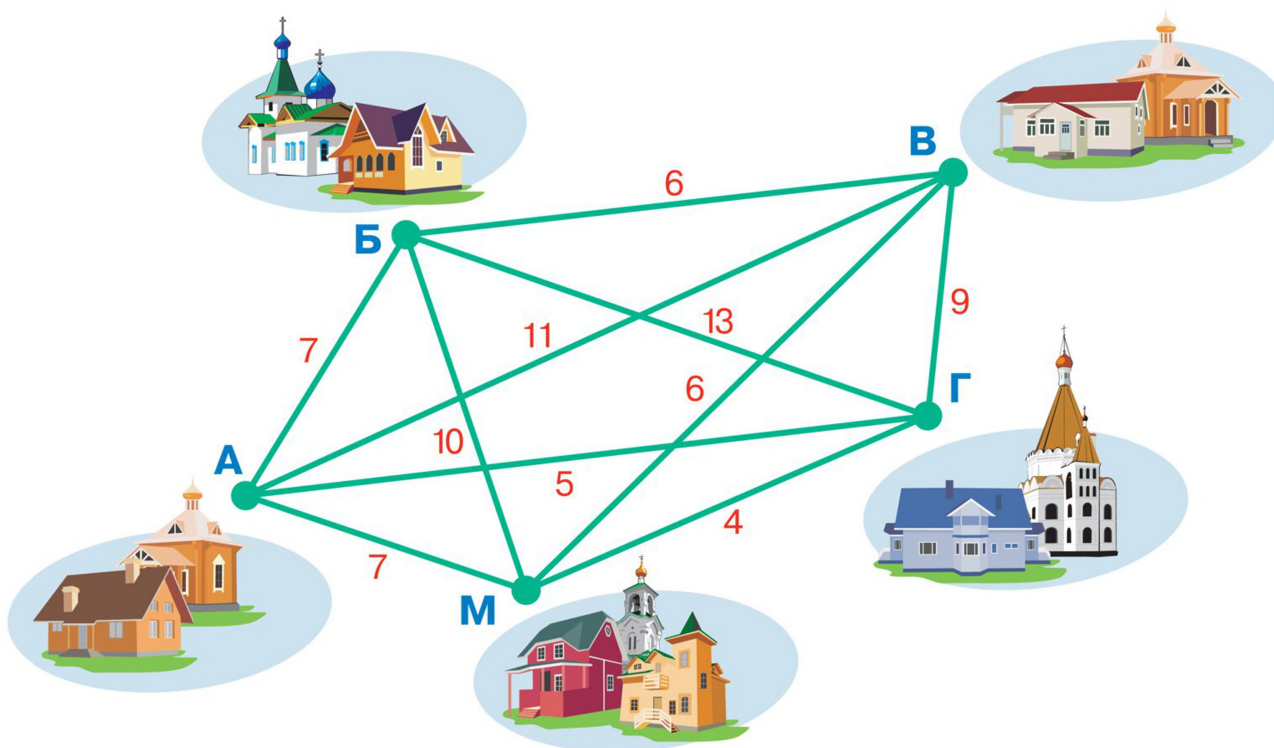


*Мороженое*



# Поиск кратчайшего пути

Рассмотри схему дорог между сёлами М, А, Б, В и Г. Числа на схеме указывают длину дорог, проложенных между сёлами, в километрах (дороги далеко не всегда бывают прямыми, поэтому длина дороги между сёлами не всегда равна самому короткому расстоянию между ними).



Такая схема называется *графом*.



Вообще **графом** называется набор точек (**вершин графа**), некоторые из которых соединены линиями (**рёбрами графа**).

Каждое дерево является графом — ведь в дереве тоже есть вершины (в них стоят элементы), и они соединены линиями — *рёбрами*.

На нашей схеме каждые две вершины соединены между собой ребром, каждое ребро *помечено числом* — длиной дороги между сёлами.

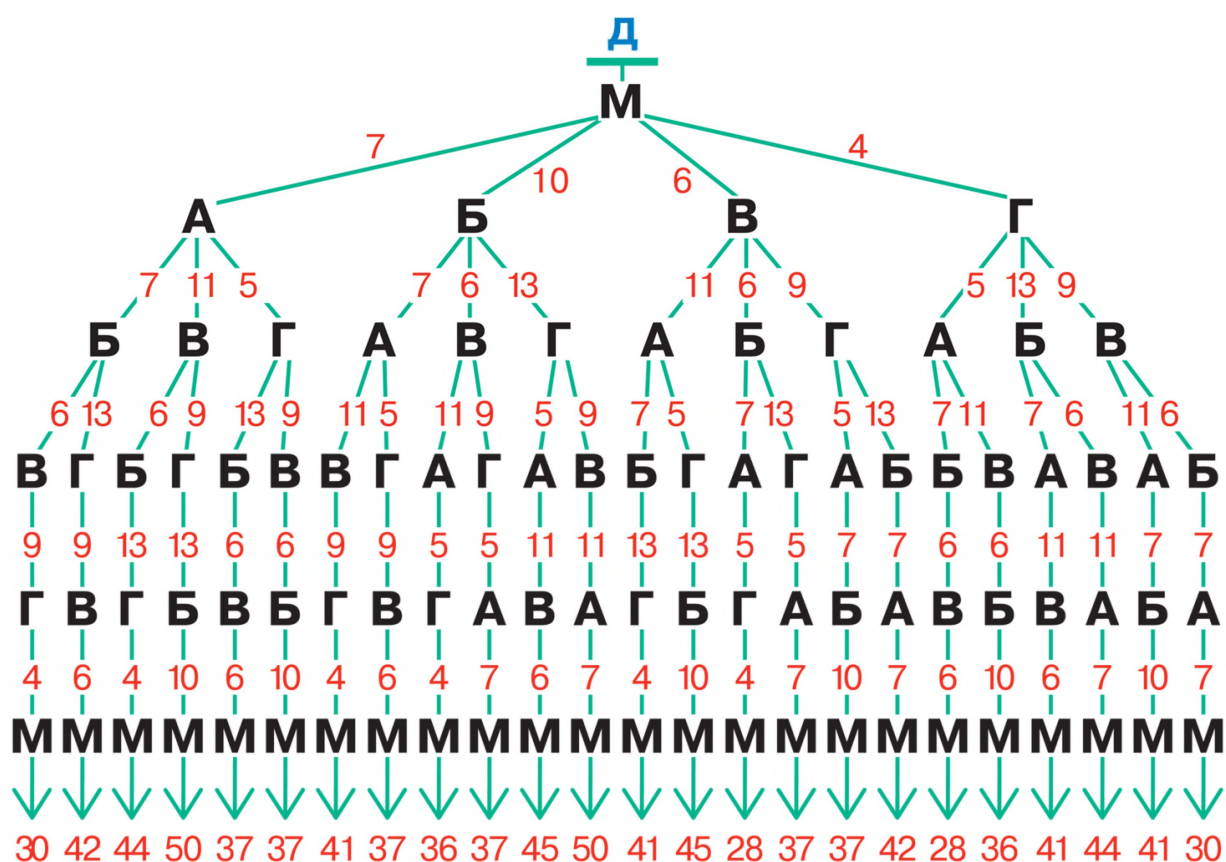
Теперь решим такую задачу:

**Задача.** В селе М находится почта, и почтальон должен развезти письма в остальные четыре села (и, конечно, вер-

нуться обратно на почту). Существует много различных *маршрутов* такой поездки. Как выбрать из них самый короткий?

Для решения задачи построим дерево  $\mathcal{D}$  перебора вариантов всех возможных маршрутов. Ясно, что если почтальон будет возвращаться в село, в котором уже побывал, то такой маршрут не будет самым коротким, поэтому такие маршруты мы рассматривать не будем.

Каждое ребро дерева  $\mathcal{D}$  пометим числом, обозначающим длину дороги между сёлами, и для каждой последовательности дерева  $\mathcal{D}$  вычислим длину маршрута (т. е. сумму чисел на его рёбрах).



Получилось всего 24 последовательности (24 варианта маршрутов). Наименьшая длина маршрута — 28 км. Этой длине пути соответствуют два маршрута:  $M \rightarrow B \rightarrow A \rightarrow G \rightarrow M$  и  $M \rightarrow G \rightarrow A \rightarrow B \rightarrow V \rightarrow M$ . Впрочем, на самом деле это один и тот же путь, только пройденный в разных направлениях.



Графы и деревья, каждому ребру которых присвоено число, называют **взвешенными**. При этом число, присвоенное ребру, называется **весом** этого ребра.



Конечно, взвешенные деревья и графы можно использовать не только для поиска кратчайшего пути. Это может быть, например, самый дешёвый путь (при передвижении на транспорте), или самый безопасный путь, или путь, который требует наименьших затрат энергии, или путь, выбранный по другим критериям. Так можно решать самые разные задачи на поиск наилучшего способа.

59

Рассмотри граф, представляющий схему дорог от дома, где живёт Коля, до школы. Нарисуй дерево перебора возможных маршрутов от дома Коли до школы. В качестве элементов такого дерева используй имена вершин графа. Через каждую точку можно проходить не больше одного раза, по дорогам со стрелками можно двигаться только в направлении стрелок. Какой путь оказался самым коротким?

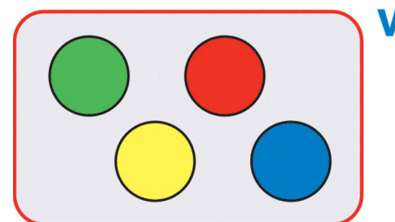


60

Построй все подмножества множества  $V$ .

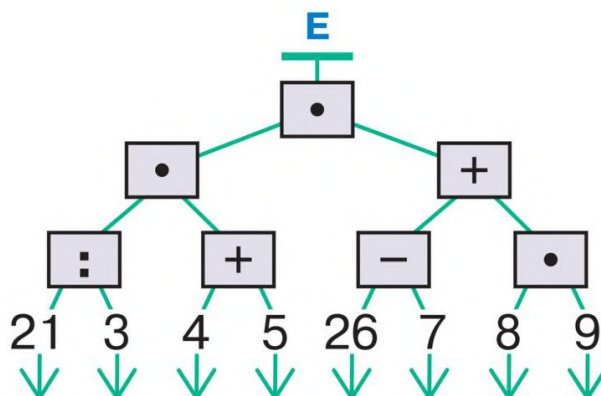
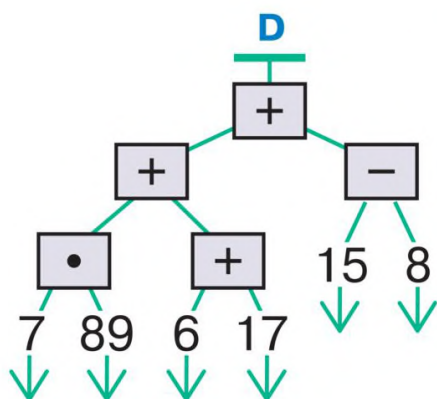
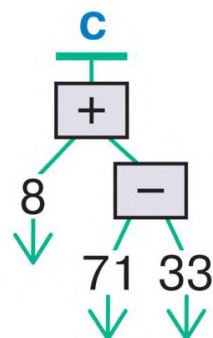
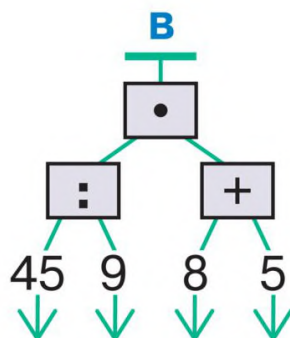
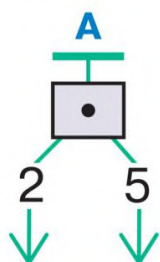


Для этого сначала построй дерево перебора подмножеств.



61

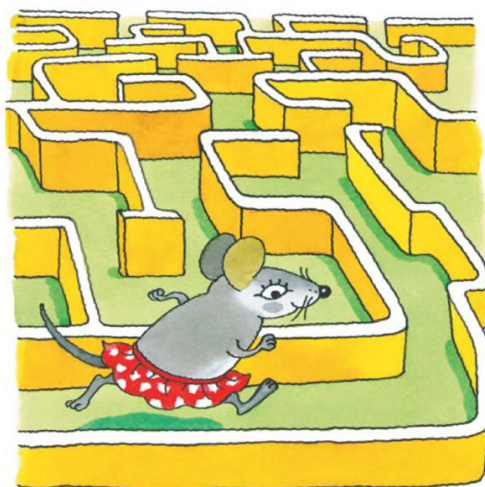
Процесс вычисления значения арифметического выражения тоже можно изобразить при помощи дерева. В таком дереве каждый *родитель* является результатом арифметической операции, применённой ко всем его *детям*. При этом *родитель* изображается в виде знака этой операции. Для каждого дерева напиши арифметическое выражение, которое соответствует этому дереву.



62

Реши задачу.

Крыса бежит по лабиринту, который устроен так, что сначала она должна выбрать один из двух проходов, затем один из трёх проходов, а за каждым из них её ожидают четыре прохода. Войдя в какой-то проход, крыса не может вернуться через него обратно. Сколькими различными путями крыса может пройти лабиринт от начала до конца?





63

Инициалы — это первые буквы имени и отчества. Например, инициалы Татьяны Михайловны — **Т. М.**, инициалы Николая Николаевича — **Н. Н.**

Сосчитай, сколько в русском языке может быть таких двухбуквенных инициалов.



Ясно, что в инициалах не встретятся буквы Ъ, Ь. Имена, начинающиеся на остальные буквы, даже если их нет в списке русских имён, могут встретиться в именах у других народов, например: Йорган, Ёшка, Щедрик. Поэтому остальные буквы следует учитывать.

Дерево перебора в этой задаче будет большим. Не строя дерева, опиши его по образцу:

1. В этом дереве ... уровня.
2. В этом дереве у каждого элемента первого уровня ровно ... детей.
3. В этом дереве всего ... листьев.

Теперь, пользуясь своим описанием, ответь на вопрос задачи.

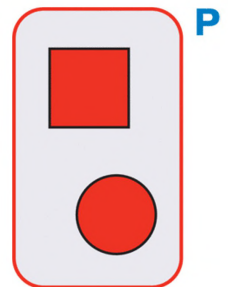
64

Построй все такие последовательности бусин, для которых все следующие утверждения истинны:

Длина этой последовательности меньше 4.

Каждая бусина этой последовательности содержится в множестве  $P$ .

В этой последовательности есть две одинаковые бусины.



65

Реши задачу.

В очереди стоят Юра, Миша, Вова, Саша и Олег. Юра стоит раньше Миши, но после Олега. Вова и Олег не стоят рядом. Саша не стоит рядом ни с Олегом, ни с Юрой, ни с Вовой. В каком порядке стоят ребята? Объясни, почему твой ответ — единственное возможное решение.

# 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** (конструкцию) — **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:

А									В

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 считается ошибкой и не приведёт к отказу при

**алг** закрасить квадрат 1

**дано** | на поле Робота нет 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** | на поле Робота нет 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**



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  $10 \times 10$ . 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 | на поле Робота нет стен,  $10 \times 10$  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
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

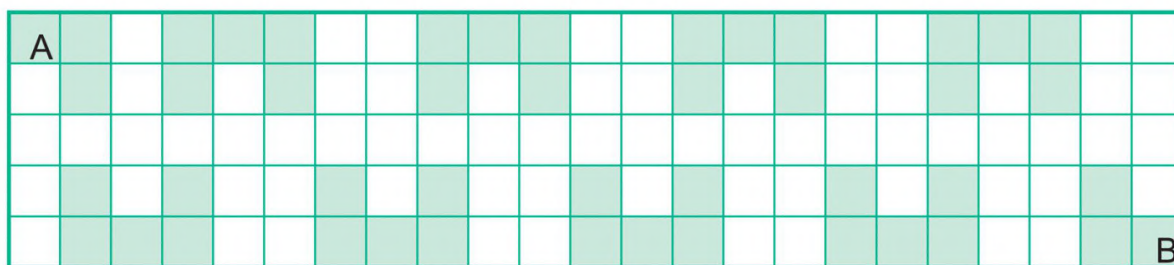
Составь алгоритм для *Кузнечика* с системой команд вперёд 3, назад 2, который заставит *Кузнечика* перейти из точки «0» в точку «100», побывав при этом в каждой точке от 0 до 100 ровно по одному разу. Используй составную команду **цикл «N раз»**.



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.



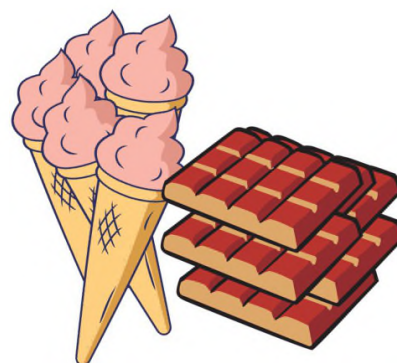
75

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

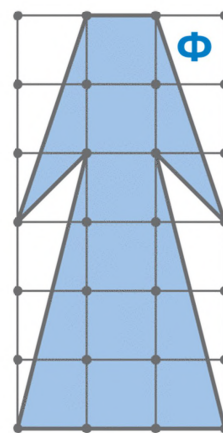
Построй алгоритм для *Удвоителя* (система команд *Удвоителя* приведена на с. 22), после выполнения которого на экране отобразится число 2014. Используй только составную команду **цикл «N раз»** и команду **прибавь 1**. Дай этому алгоритму имя **прибавь 2014 раз по 1**. Сколько всего команд компьютер 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

Найди площадь многоугольника Ф.

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



# Повторение

82

Выполнив алгоритм *Маршрут*, в котором не больше 10 команд (без вспомогательных алгоритмов и циклов) и нет команды *закрась*, *Робот* из клетки А попал в клетку В.

а) Каков результат выполнения этого же алгоритма, если начальное положение *Робота* — клетка С? клетка D? клетка Е?

б) Может ли в этом алгоритме быть ровно 10 команд? ровно 8 команд?

					В	
С	А					
Е			Д			

83

Пользуясь заполненной частью таблицы, заполни остальные её столбцы, имея в виду то, что итальянский и французский языки произошли от латинского.

Французский

mardi, vendredi,  
dimanche, lundi,  
samedi, jeudi,  
mercredi

Итальянский

venerdi, domenica,  
lunedì, sabato,  
giovedì, mercoledì,  
martedì

Русский	Латинский	Французский	Итальянский
понедельник	Lunae dies		
вторник	Martis dies		
среда	Mercurii dies		
четверг	Jovis dies		
пятница	Veneris dies		
суббота	Saturni dies		
воскресенье	Solis dies		

84

Схема линий метро — это тоже граф, вершинами которого являются станции, а рёбрами — линии между станциями. Пользуясь этой схемой линий метрополитена Санкт-Петербурга, выпиши названия всех станций, до которых можно доехать от станции «Новочеркасская» не больше чем за 11 мин, если на переезд между каждыми двумя соседними станциями уходит 3 мин, а на каждую пересадку — 5 мин.



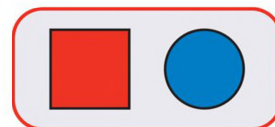
Для решения построй дерево перебора вариантов, присвоив каждому ребру свой вес — время в пути.





85

Построй все такие последовательности бусин, для каждой из которых следующие утверждения истинны:



Длина этой последовательности — 2.

Каждая бусина этой последовательности есть в множестве К.

86

В настоящее время в России принята система автомобильных номеров, в которых сначала идёт буква русского алфавита, потом три цифры, а затем ещё две буквы. Кроме того, отдельно справа пишется номер региона, к которому приписан автомобиль. Сосчитай, сколько существует разных автомобильных номеров в одном регионе (т. е. если не учитывать номер региона).



Следует учесть, что в автомобильных номерах буквы Ё, Й, Ъ, Ы, Ь не используются.

Дерево перебора в этой задаче будет большим. Не строя дерева, опиши его по образцу:

1. В этом дереве всего ... уровней.
2. В этом дереве всего ... элементов первого уровня.
3. В этом дереве у каждого элемента первого уровня ровно ... детей.
4. В этом дереве у каждого элемента второго уровня ровно ... детей.

.....

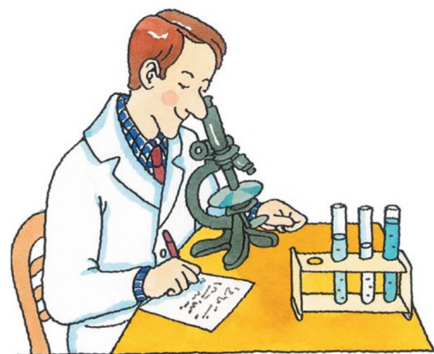
В этом дереве всего ... листьев.

Пользуясь своим описанием, ответь на вопрос задачи.

87

Реши задачу.

Один биолог открыл удивительную разновидность амёб. Каждая из них через одну минуту делится на две. Биолог кладёт амёбу в пробирку, и ровно через час пробирка оказывается заполненной амёбами. Сколько времени потребуется, чтобы вся пробирка заполнилась амёбами, если в неё положить не одну, а две амёбы?



88

Нарисуй результат выполнения *Роботом* алгоритма *узор3*. Сколько раз *Робот* выполнил команду закрасить, выполняя алгоритм *узор3*? Сколько всего на поле стало закрашенных клеток?

89

При выполнении вспомогательного алгоритма *узор4* *Робот* закрашивает некоторые клетки в квадрате  $4 \times 4$ , причём путь *Робота* при выполнении этого алгоритма начинается и заканчивается в левом верхнем углу этого квадрата. Составь алгоритм, который закрашивает 25 одинаковых фрагментов  $4 \times 4$  в квадрате  $20 \times 20$  клеток таким же узором. Используй составную команду **цикл «N раз»**.

90

*Кузнечик* с системой команд *вперёд 3*, *назад 2* выполнил некоторый алгоритм из 5 команд (без вспомогательных алгоритмов и циклов) и оказался в той же точке числовой прямой, с которой он начал выполнять алгоритм. Напиши такую программу. Сколько различных программ с таким свойством существует?

91

Реши задачу.  
В классе 35 учеников. Каждый из них занимается в кружках: в биологическом — 17 человек, в литературном — 30, в математическом — 13. Сколько из этих учеников занимается только в одном кружке, если известно, что пятеро из них занимается во всех трёх кружках?

алг узор3

**дано** | на поле нет  
| стен,  
|  $10 \times 10$ ,  
| Робот в  
| верхнем  
| правом  
| углу

**надо** |**нач**

**нц** 9 **раз**  
закрасить  
вниз

**кц**

**нц** 8 **раз**  
закрасить  
влево

**кц**

**нц** 7 **раз**  
закрасить  
вверх

**кц**

**нц** 6 **раз**  
закрасить  
вправо

**кц**

**нц** 5 **раз**  
закрасить  
вниз

**кц**

**нц** 4 **раз**  
закрасить  
влево

**кц**

**нц** 3 **раз**  
закрасить  
вверх

**кц**

**нц** 3 **раз**  
закрасить  
вправо

**кц****кон**



92



Дано множество Р названий дней недели на трёх языках — латышском, литовском и эстонском. Используя только данный словарь, найди переводы всех слов и заполни таблицу. Можно вырезать таблицу и слова из листа вырезания и составить таблицу в тетради.

Р

Antradienis  
Ceturtdiena  
Esmaspäev  
Ketvirtadienis  
Kolmapäev  
Laupäev  
Otrdiena  
Neljäpäev  
Penktadienis  
Piektadiena  
Pirmadienis  
Pirmdiena  
Pühapäev  
Reede  
Sekmadienis  
Šeštadienis  
Sestdiena  
Svētdiena  
Teisipäev  
Trečadienis  
Trešdiena

Otrdiena (лат.) — Antradienis (лит.)  
Sekmadienis (лит.) — Pühapäev (эст.)  
Neljäpäev (эст.) — четверг  
Reede (эст.) — Penktadienis (лит.)  
Ketvirtadienis (лит.) — Ceturtdiena (лат.)  
Antradienis (лит.) — Teisipäev (эст.)  
Trešdiena (лат.) — Kolmapäev (эст.)  
Esmaspäev (эст.) — Pirmadienis (лит.)  
Trečadienis (лит.) — среда  
Šeštadienis (лит.) — Laupäev (эст.)  
Sestdiena (лат.) — суббота  
Pühapäev (эст.) — воскресенье  
Penktadienis (лит.) — Piektadiena (лат.)  
Teisipäev (эст.) — вторник  
Svētdiena (лат.) — Sekmadienis (лит.)  
Pirmadienis (лит.) — понедельник

Русский	Латышский	Литовский	Эстонский
понедельник			
вторник			
среда			
четверг			
пятница			
суббота			
воскресенье			

# Игры с полной информацией

Игры — важная часть жизни детей. Взрослые тоже часто играют в игры; кроме того, изучению игр посвящены большие разделы информатики и математики.

Существует много разных видов игр: спортивные, компьютерные... Игры, которые мы будем изучать, в математике называются *играми с полной информацией*. В таких играх в любой момент каждому игроку известно всё, что произошло в игре к этому моменту и какие ходы возможны. Это, например, такие игры, как шахматы, шашки, крестики-нолики. Игры, исход которых хотя бы частично зависит от случая, например большинство карточных игр, игральные кости, морской бой, не являются играми с полной информацией.

Мы будем использовать следующие понятия:

*Игроки* — это участники игры. В наших играх всегда будет два игрока, мы будем называть их Первый и Второй.

*Позиция игры* — это то, что в какой-то момент получилось при игре двух игроков. Например, в шашках или шахматах позиция игры — это расстановка фигур на поле после хода одного из игроков. В игре крестики-нолики — то, как расставлены крестики и нолики в этот момент игры.

*Ход игры* изменяет позицию. Игроки делают ходы поочерёдно. Первый ход всегда делает Первый, второй ход делает Второй, третий ход — снова Первый, четвёртый — снова Второй и т. д.

*Начальная позиция* — это позиция, которая имеется перед началом игры.

*Заключительная позиция* — такая ситуация в игре, после которой игра продолжаться не может.

*Партия игры* — одна игра двух игроков от начала (начальной позиции) и до выигрыша одного из игроков или ничьей. Партию игры можно изобразить в виде последовательности всех позиций, которые получились в этой партии: начиная с начальной позиции до заключительной. Кто из игроков выиграл в этой партии, определяется по тому, какой именно заключительной позицией закончилась партия и кто из игроков сделал последний ход.



### Описание правил игры:

- описывают все возможные *начальные позиции*;
- определяют, какие *ходы* возможны в каждой позиции;
- описывают все возможные *заключительные позиции* и для каждой заключительной позиции указывают, каков *исход партии* — кто выиграл или партия закончилась вничью.

Мы начнём изучать игры с полной информацией с игры *Камешки*.

### Правила игры *Камешки*

Начальная позиция. Куча камешков (5 штук и больше). Вместо камешков можно использовать любые предметы — пуговицы, фасоль, палочки, скрепки.

Возможные ходы. Игрок забирает из кучи разрешённое число камешков. Сколько именно камешков разрешается убрать, определяется дополнительными правилами игры.

Как определить победителя. Заключительная позиция игры — это пустая куча камешков. При этом игрок, который забрал последний камешек из кучи, считается выигравшим (и, значит, в этой игре ничьих не бывает).

Как видно из правил, в игре *Камешки* неважно, какие именно камешки были в куче до начала игры и какие именно камешки забирает игрок на своём ходу. Важно только то, сколько камешков было до начала и сколько их осталось после каждого хода.

Рассмотрите *последовательность W позиций* партии игры *Камешки* (начальная позиция — 6 камешков, разрешается брать 1 или 2 камешка за один ход):

w | 6-4-3-2-0 ➔

Позиции, которые получились после ходов Первого, помечены в последовательности W синим цветом. Позиции, которые получились после ходов Второго, остались чёрными.

Первый член последовательности W — это начальная позиция, второй член — позиция после первого хода (напомним, что

первый ход всегда делает Первый) и т. д., последний член последовательности  $W$  — заключительная позиция.

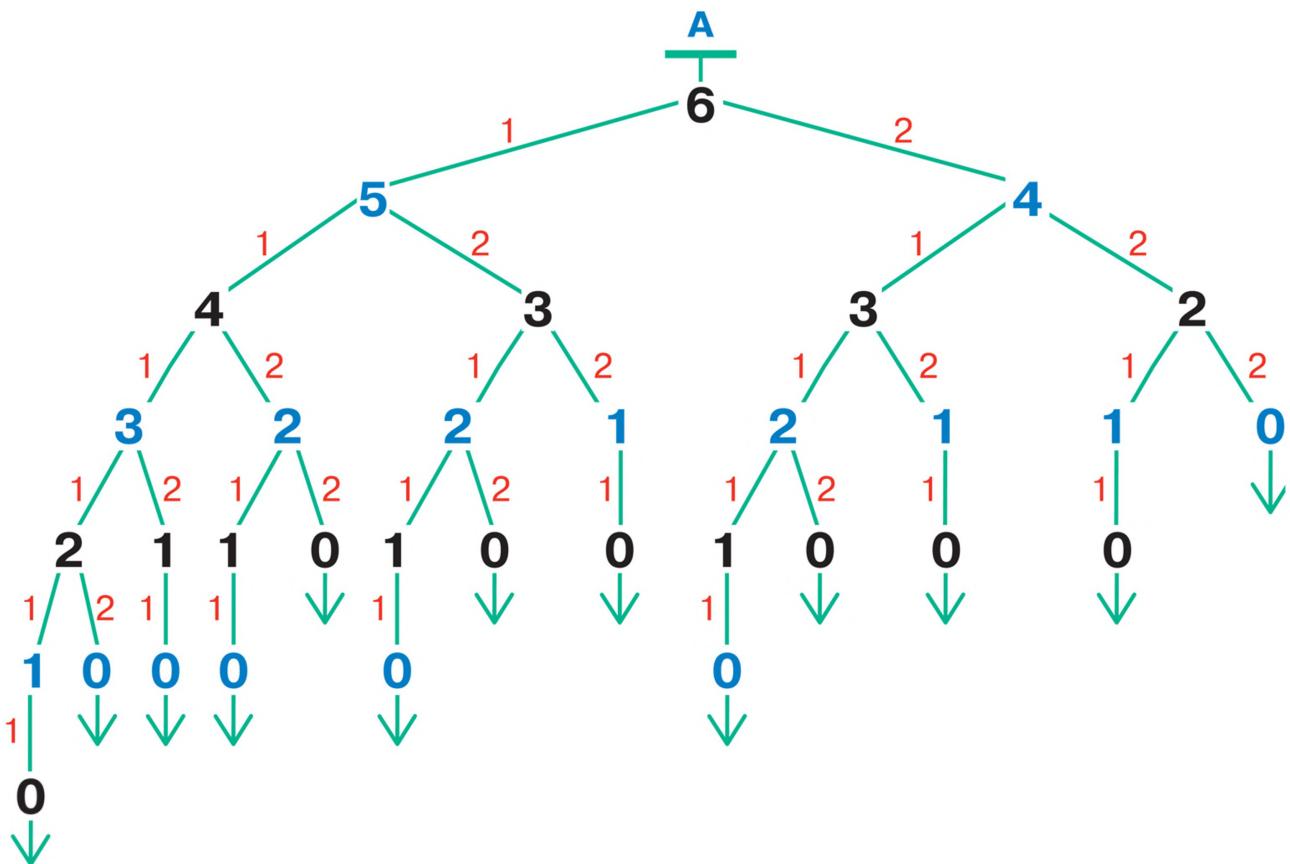
Заключительная позиция получилась в результате хода Второго: Второй забрал 2 оставшихся камешка и выиграл в партии  $W$ .

## Дерево игры

Последовательность позиций изображает только одну партию. Построим теперь *дерево перебора всех возможных партий игры* с данными правилами. Такое дерево называется *деревом игры*.

Единственным элементом первого уровня дерева игры является начальная позиция, элементами второго уровня — все позиции, возможные после первого хода, элементами третьего уровня — все позиции, возможные после второго хода, и т. д. Листья дерева игры — это заключительные позиции.

Позиции нашей игры *Камешки* после ходов Первого помечены в дереве  $A$  синим, а после ходов Второго — чёрным. Для удобства можно пометить каждое ребро дерева числом, которое показывает, сколько камешков забрал игрок на этом ходу.







Каждая последовательность дерева игры является последовательностью позиций одной из возможных партий игры. Последовательности дерева — это возможные в данной игре партии. При этом в дереве есть последовательность для каждой возможной партии.

**93**

Игроки играют в игру *Камешки* с такими правилами: начальная позиция — 11 камешков; за один ход можно взять **1**, **2** или **3** камешка. Придумай две партии этой игры так, чтобы в одной партии выиграл Первый, а в другой — Второй. Построй последовательности позиций для этих партий. Позиции изображай числами.

**94**

Устройте соревнование с соседом по парте — сыграйте 8 партий в *Камешки* (начальная позиция **7** камешков, разрешается брать **1** или **2** камешка за один ход).



Начинайте игру по очереди: пусть один из вас играет Первым в партиях с чётными номерами, а другой — с нечётными. Заполни таблицу соревнования (образец таблицы дан справа, такая же таблица есть на вкладыше тетради проектов). В клетки таблицы записывай, сколько очков набрал каждый игрок в очередной партии. За каждую победу игрок получает 1 очко, а за поражение — 0 очков. Сколько раз в вашем соревновании выиграл Первый?

Игроки Партии	Твоё имя	Имя соседа
1-я партия		
2-я партия		
3-я партия		
4-я партия		
5-я партия		
6-я партия		
7-я партия		
8-я партия		
Итого		

**95**

Нарисуй дерево Б игры *Камешки* (начальная позиция **5** камешков, разрешается брать **1** или **2** камешка за один ход). Запиши позиции, которые получились в результате хода Первого, синим, а позиции, которые получились в результате хода Второго, чёрным. Обведи синим листья дерева Б, соответствующие партиям, в которых выиграл Первый. Обведи зелёным листья, соответствующие партиям, в которых выиграл Второй.

96

Составь алгоритм, который заставит *Робота* закрасить клетки квадратного поля  $10 \times 10$  в шахматном порядке, начиная с левого верхнего угла поля. Используй составную команду **цикл «N раз»** и вложенный цикл.

97

Пусть  $S$  — множество всех букв, которые встречаются в слове **КАРТОШКА**, и  $R$  — множество всех букв, которые встречаются в слове **МАКАРОНЫ**. Построй пересечение и объединение множеств  $S$  и  $R$ .

98

Построй последовательность  $A$  позиций такой партии игры *Камешки* (начальная позиция **5** камешков, разрешается брать **1**, **2** или **4** камешка за один ход), в которой на третьем ходу партии выиграл Первый.

Построй последовательность  $B$  позиций такой партии той же игры, в которой на четвёртом ходу выиграл Второй. Затем построй дерево  $B$  этой игры. Найди в дереве  $B$  последовательность, равную последовательности  $A$ , и обведи лист, соответствующий этой последовательности, красным. Найди в дереве  $B$  последовательность, равную последовательности  $B$ , и обведи лист, соответствующий этой последовательности, синим.

99

Прочитай описание игры *Крестики-нолики*.

Игра ведётся на поле размером  $3 \times 3$  клетки. В игре принимают участие два игрока, которые делают ходы по очереди. Во время хода игрок рисует свой значок в свободной клетке поля. Первый игрок рисует крестики, второй игрок рисует нолики. Если на поле возник ряд из трёх крестиков (по горизонтали, по вертикали или по диагонали), то выиграл первый игрок; если возник ряд из трёх ноликов, то выиграл второй игрок. Если все клетки поля заполнены значками, но ряда из трёх одинаковых значков не возникло, то игра закончилась вничью.



Сформулируй правила этой игры по образцу:

Начальная позиция.

Возможные ходы.

Как определить победителя. В этой игре заключительные позиции бывают трёх видов:

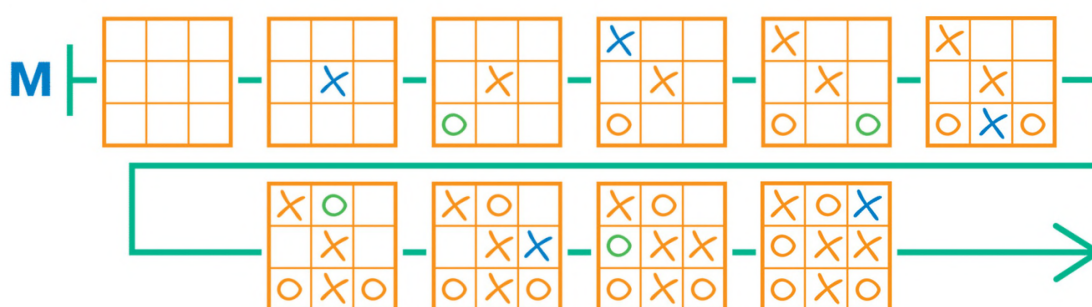
- 1) ... (выиграл Первый);
- 2) ... (выиграл Второй);
- 3) ... (партия закончилась вничью).

Нарисуй примеры заключительных позиций каждого из трёх видов, напиши, кто выигрывает при такой заключительной позиции.

100

Рассмотри последовательность М позиций партии игры Крестики-нолики. Ответь на вопросы:

- а) Сколько ходов было сделано в этой партии?
- б) Сколько ходов сделал Первый и сколько — Второй?
- в) Каков исход партии?



101

Реши лингвистическую задачу.

Даны болгарские слова и словосочетания и их переводы на русский язык в перепутанном порядке:

ледена планина  
иглолистно дърво  
горска ягода  
горски пояс  
иглолистна гора  
градина  
планинска верига  
градинска ягода  
горолом

сад  
горная цепь  
хвойный лес  
бурелом  
клубника  
земляника  
лесополоса  
айсберг  
хвойное дерево

Установи русские переводы всех болгарских слов и словосочетаний.



Болгарский и русский языки — родственные, и многие болгарские слова похожи на русские. Например, болгарское слово «дърво» похоже на русское слово «дерево» и действительно переводится как «дерево». Но похожие русские и болгарские слова необязательно обозначают одно и то же. Поэтому для решения задачи нужно провести исследование всех слов и рассуждение.

## Команды-запросы *Робота*. Условие

Представьте себе начальника, который отдаёт приказы, но не получает никаких донесений о результатах их выполнения; повара, который не может попробовать блюдо; шофёра, ведущего автомобиль с закрытыми глазами. Понятно, что так далеко не уедешь: если мы хотим составлять алгоритмы для решения сложных задач, то надо уметь не только командовать исполнителем, но и **анализировать обстановку**, в которой он оказался.

Например, для *Робота* **обстановка** — это поле, на котором он находится (расположение стен на поле, раскраска клеток, температура и уровень радиации в каждой клетке), а также то, в какой клетке находится *Робот*.

Для анализа обстановки у *Робота* есть 12 команд-запросов, будем для краткости называть их просто **запросами**:

### Команды-запросы *Робота*

сверху стена  
снизу стена  
справа стена  
слева стена  
сверху свободно  
снизу свободно  
справа свободно  
слева свободно  
клетка закрашена  
клетка чистая  
температура  
радиация



Отвечая на запрос, *Робот* сообщает информацию об обстановке в той клетке, где он находится: о закраске этой клетки, о наличии стен на границах клетки, о температуре и радиации в клетке.

Смысл этих вопросов ясен из их имён. Мы будем использовать пока только первые 10 команд-запросов — с командами-запросами температура и радиация мы будем работать позже.

Итак, рассмотрим команды-запросы, которые касаются наличия или отсутствия стены на границах клетки и того, закрашена ли клетка, на которой стоит *Робот*. Когда *Робот* получает такую команду, он определяет истинность соответствующего утверждения и даёт ответ — «истина» или «ложь». Можно считать, что, давая такую команду-запрос, компьютер или человек задаёт *Роботу* вопрос и получает от него в ответ «да» или «нет».

В нижней части пульта дистанционного управления *Роботом* (см. рисунок слева) есть кнопки «Стена/Закрашено», «Свободно/Чисто». Например, чтобы дать *Роботу* команду-запрос сверху стена, надо нажать на пульте две кнопки: сначала кнопку «Стена», а потом кнопку со стрелкой вверх. Чтобы дать *Роботу* команду-запрос клетка чистая, надо нажать на пульте кнопку «Чисто», а потом центральную кнопку. Для получения ответов *Робота* используется экран пульта — на нём появляется слово «да» («истина») или «нет» («ложь»).



Задача 1. *Робот* охраняет помещение, состоящее из двух соседних клеток (см. рисунок справа). Неизвестно, в какой из двух клеток находится *Робот*. Необходимо перевести его в другую клетку.



Сначала решим задачу в режиме непосредственного управления *Роботом*. По условию задачи мы не видим, в какой клетке поля он находится. Чтобы узнать, в какой клетке *Робот* сейчас находится, дадим ему команду-запрос: справа свободно. Ответ «истина» («да») означает, что *Робот* в левой клетке и должен сделать шаг вправо. Ответ «ложь» («нет») означает, что *Робот* в правой клетке и должен сдвинуться влево. Наши действия при непосредственном управлении *Роботом* можно описать так: *если* справа свободно, *то* скомандовать вправо, *иначе* (если справа не свободно, стоит стена) скомандовать влево.

При решении задачи 1 мы давали *Роботу* запрос и в зависимости от ответа выбирали следующее действие. Для описания такого выбора в алгоритмическом языке существует специальная *составная команда если*. Общий вид команды *если* таков:

```
если <условие>
  то
    <последовательность команд 1>
  иначе
    <последовательность команд 2>
все
```

или

```
если <условие>
  то
    <последовательность команд 1>
все
```

Служебные слова **если**, **то**, **иначе** имеют тот же смысл, что и в русском языке. Слово **все** означает конец команды («всё»), это слово пишется строго под словом **если**. После **то** записывается последовательность команд (последовательность команд 1), которая выполняется в случае истинности условия. После **иначе** записывается другая последовательность команд (последовательность команд 2), которая выполняется в случае ложности условия. Последовательность команд 2 вместе со служебным словом **иначе** может отсутствовать.

При обработке составной команды **если** компьютер посылает *Роботу* команду-запрос, чтобы проверить истинность условия,



записанного в строке **если**. Если *Робот* сообщает, что условие истинно, то компьютер посылает *Роботу* команды из последовательности 1, а команды из последовательности 2 пропускает. Если *Робот* сообщает, что условие ложно, то компьютер посылает *Роботу* команды из последовательности 2 (если она есть). Если условие ложно, а последовательность команд 2 вместе с **иначе** отсутствует, то компьютер сразу переходит к выполнению команд, записанных после слова **все**.

Вот как можно записать алгоритм решения задачи 1:

```

алг дежурство
    дано | Робот в одной из клеток
         | «двухкомнатного» помещения
    надо | Робот в другой клетке
нач
    если справа свободно
        то вправо
    иначе влево
все
кон

```

**Задача 2.** Робот стоит в левом конце горизонтального коридора длиной в 23 клетки. Нижняя стена коридора сплошная, а в верхней имеется несколько выходов. Надо перевести Робота в правый конец коридора (из А в В) и закрасить все те клетки коридора, из которых есть выход вверх (см. рисунок). Количество выходов и их точное расположение заранее неизвестны.

[illegible]

Итак, *Робот* должен пройти по горизонтальному коридору, при этом закрашивать надо только те клетки коридора, где есть выход вверх. Другими словами, если сверху свободно, то клетку надо закрасить, иначе — красить не надо. Попробуем записать это в виде составной команды **если**:

если сверху свободно  
то закрасить  
иначе ...  
все

А что собственно «иначе»? Как записать, что иначе ничего делать не надо? Для таких случаев и существует краткая форма составной команды **если**, в которой слова **иначе** вообще нет:

```
если сверху свободно  
    то закрасить  
все
```

При выполнении этой составной команды компьютер даст *Роботу* команду определить истинность условия сверху свободно. Если *Робот* ответит «истинно», то компьютер скомандует *Роботу* закрасить. Если же *Робот* ответит «ложно», то компьютер вызов команды закрасить пропустит. Вот алгоритм решения задачи 2:

```
алг разметка выходов из коридора  
  дано | Робот в левой клетке  
        | горизонтального коридора  
  надо | Робот в самой правой клетке коридора,  
        | клетки коридора, из которых  
        | есть выход вверх, закрашены  
нач  
  если сверху свободно  
    то закрасить  
  все  
  нц 22 раз  
    вправо  
    если сверху свободно  
      то закрасить  
    все  
  кц  
кон
```

Обрати внимание: в этом алгоритме первая составная команда **если** служит для того, чтобы *Робот* раскрасил клетку А, если из неё есть выход из коридора. Дальше *Робот* 22 раза повторяет следующее: делает шаг вправо, проверяет, свободно ли сверху, и закрашивает клетку, если свободно. Если сверху стоит стена, *Робот* клетку не закрашивает, а сразу переходит на шаг вправо.



102

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

A			B					C
				D				
			E			F		
G							H	

Клетка	A	B	C	D	E	F	G	H
Утверждение								
сверху свободно								
справа стена								
снизу свободно								
слева стена								
клетка чистая								

103

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


Клетка	K	L	M	N	O	P	Q	R
Утверждение								
сверху свободно	И	Л	Л	И	Л	И	Л	Л
справа стена	Л	И	Л	Л	Л	И	Л	Л
снизу свободно	Л	Л	И	Л	Л	И	И	Л
слева стена	Л	Л	Л	И	Л	Л	И	И
клетка чистая	И	Л	Л	Л	И	Л	Л	И

104

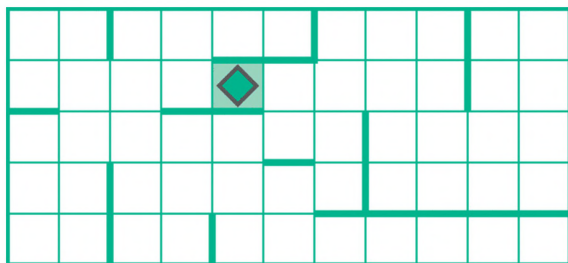
Вот буквы армянского алфавита. Найди три одинаковые буквы, напиши такую букву в тетради.

Է Գ Ի Լ Դ Զ Ը Ժ Լ Խ Բ Գ Դ Ը  
 Ն Ս Ր Պ Ւ Վ Ն Ն Ն Ն Ն Ն Ն Ն  
 Ի Ս Ն Ն Ի Ի Ր Ս Ի Լ Զ Է Գ

105



Нарисуй состояние *Робота* после выполнения им алгоритма переход из данного начального состояния.



106



Построй такую последовательность партии игры *Крестики-нолики*, в которой на седьмом ходу выигрывает Первый.

107

Реши задачу.

Разложи по семи кошелькам 127 рублёвых монет так, чтобы любую сумму от 1 до 127 р. можно было выдать, не открывая кошельков (другими словами, отдав всё содержимое одного или нескольких кошельков).

```

алг переход
  дано |
  надо |
  нач
    влево
    влево
  если снизу
    свободно
    то вниз
    вправо
    вправо
  все
    вправо
    закрась
    вправо
    закрась
    вниз
  если снизу
    свободно
    то вниз
    иначе влево
    вниз
  все
кон
  
```

108

Построй состояние *Робота* после выполнения каждого алгоритма из каждого из данных начальных состояний. Построй

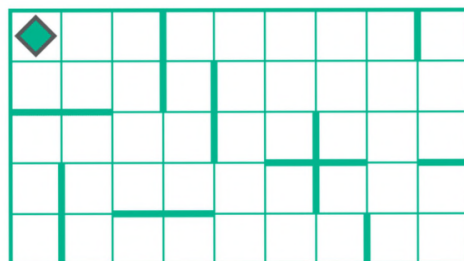


таблицу по образцу — в каждую клетку наклейте заготовку поля, раскрасьте клетки, нарисуйте положение *Робота*.

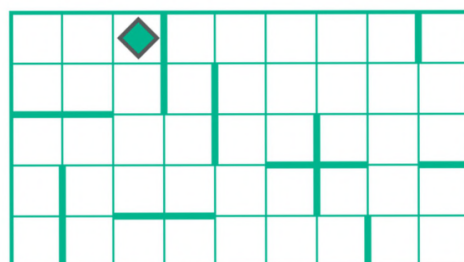
```

алг А
  дано |
  надо |
нач
  нц 6 раз
    если справа свободно
      то вправо
    все
    если снизу свободно
      то вниз
    все
  кц
кон
  
```

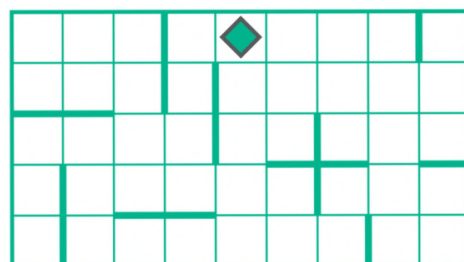
Начальное состояние 1:



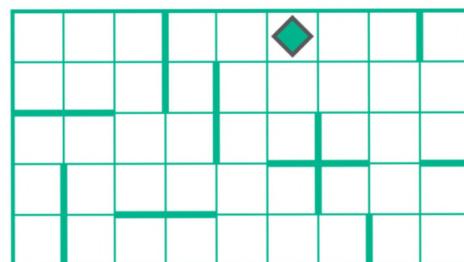
Начальное состояние 2:



Начальное состояние 3:



Начальное состояние 4:



```

алг Б
  дано |
  надо |
нач
  нц 6 раз
    если снизу свободно
      то вниз
    все
    если справа свободно
      то вправо
    все
  кц
кон
  
```

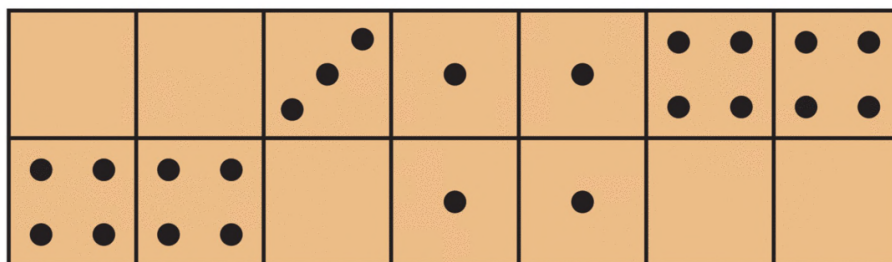
Начальное состояние	Состояние после выполнения алгоритма А	Состояние после выполнения алгоритма Б

109

Несколько костяшек из одного набора домино уложили так, как показано на рисунке. Определи, где проходят границы между костяшками. Для этого вырежи такой же рисунок со вкладыша тетради проектов, наклей его в тетрадь и проведи на нём эти границы жирными линиями.



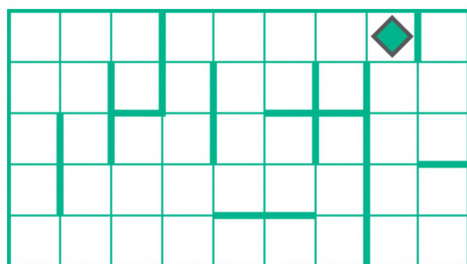
Нужно иметь в виду, что в одном наборе домино все костяшки разные.



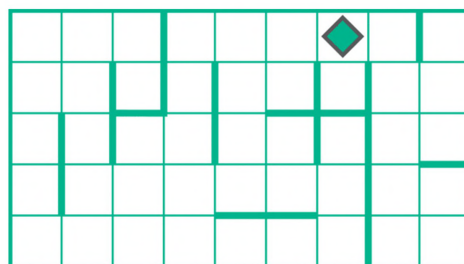
110

Составь такой алгоритм для *Робота*, после выполнения которого из каждого из данных начальных состояний *Робот* окажется в левом нижнем углу поля.

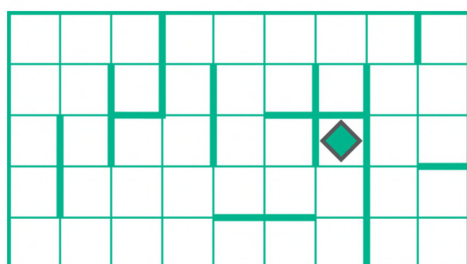
Начальное состояние 1:



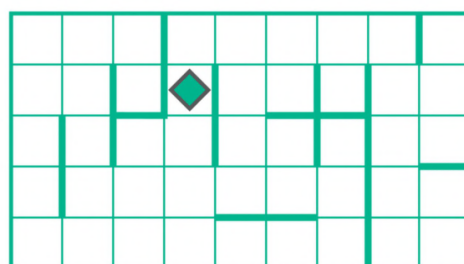
Начальное состояние 2:



Начальное состояние 3:



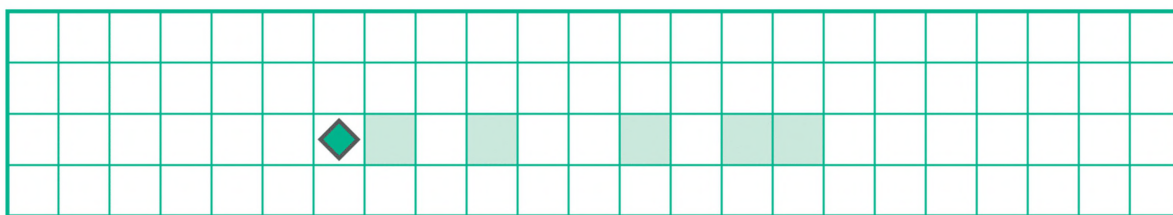
Начальное состояние 4:



111

На поле *Робота* стен нет. В ряду правее той клетки, где стоит *Робот*, некоторые клетки закрашены, причём самая дальняя закрашенная клетка находится не дальше чем в 10 шагах от *Робота*, например, как на рисунке ниже. Какие именно клетки закрашены изначально, неизвестно, известно только, что больше закрашенных клеток на поле нет.





Составь алгоритмы, которые заставляют *Робота*:

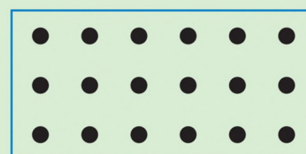
- а) закрасить клетку на шаг ниже каждой из закрашенных изначально;
- б) закрасить клетку на шаг выше и на шаг ниже каждой из закрашенных изначально;
- в) закрасить клетку на шаг левее каждой из закрашенных изначально;
- г) закрасить клетку на шаг правее каждой из закрашенных изначально.

112

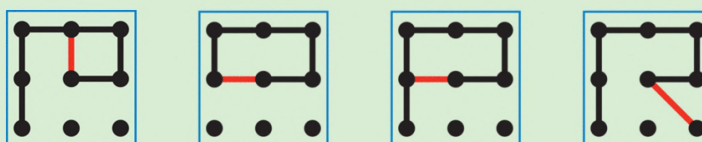
Даны правила игры *Ползунок*.

### Правила игры *Ползунок*

Начальная позиция. Игровое поле состоит из нескольких рядов точек, расположенных в прямоугольнике. Пример такого поля приведён справа.



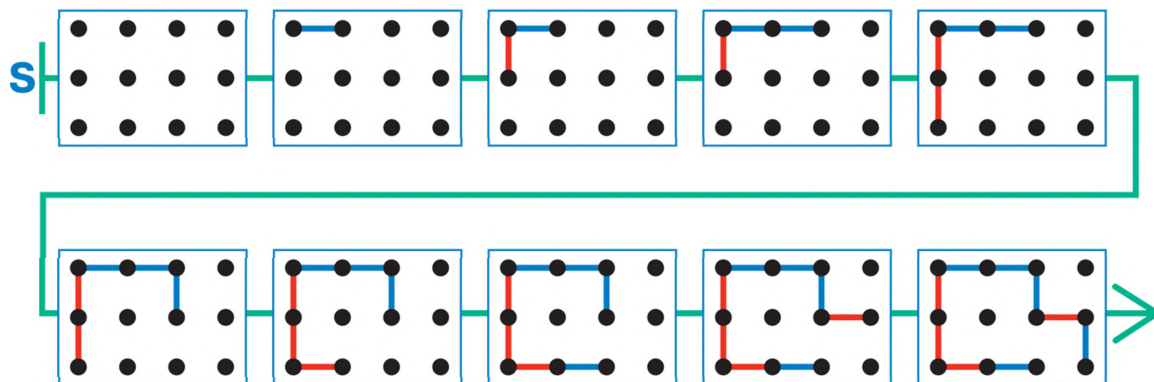
Возможные ходы. На первом ходу Первый соединяет горизонтальным или вертикальным отрезком две любые соседние точки. На каждом из следующих ходов игрок проводит горизонтальный или вертикальный отрезок, соединяющий один из концов получившейся до этого ломаной линии с какой-нибудь соседней точкой, через которую линия ещё не прошла. Например, такие ходы не разрешены:



Как определить победителя. Игра заканчивается, если очередной ход сделать нельзя. Выигрывает игрок, который сделал последний ход.

Рассмотри последовательность S позиций партии игры *Ползунок* на поле  $4 \times 3$  точки. Ходы Первого помечены синим, ходы Второго — оранжевым. Ответь на вопросы:

- Сколько ходов было сделано в этой партии?
- Сколько ходов сделал Первый и сколько — Второй?
- Кто выиграл?



113

Придумай такую партию игры *Ползунок* на поле  $4 \times 3$ , в которой выиграл Второй. Построй последовательность позиций этой партии. Заготовки полей есть на вкладыше тетради проектов. Теперь построй последовательность P позиций такой партии, в которой выиграл Первый.

114

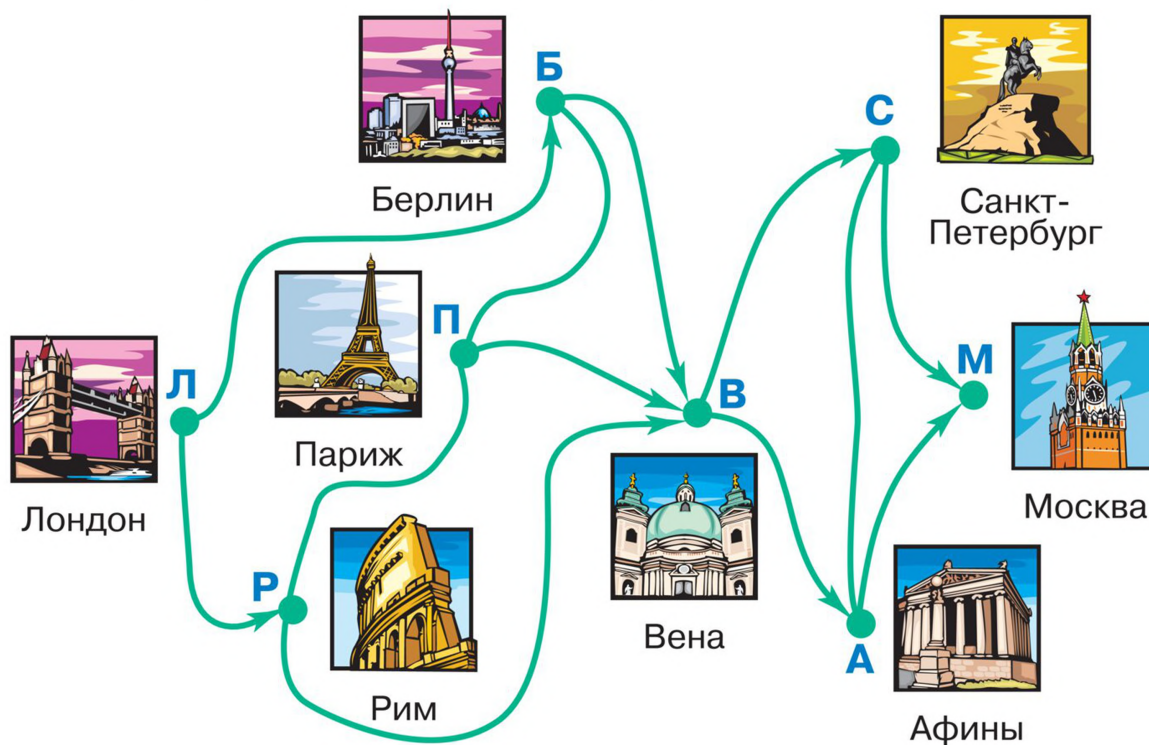
Построй дерево, для которого все утверждения в таблице имеют указанные истинностные значения.

Имя	Утверждение	Значение
A	Все листья этого дерева — элементы третьего уровня.	И
B	Все бусины в этом дереве только двух цветов — чёрные и белые.	И
C	В этом дереве есть две одинаковые последовательности.	Л
D	Каждый элемент этого дерева — круглая бусина.	И
E	В этом дереве меньше восьми последовательностей.	Л



115

Определи по графу, сколькими способами можно проехать из города Л в город М. По дорогам со стрелками можно двигаться только в направлении стрелок, по каждой дороге можно проезжать не больше одного раза.



## Выигрышная стратегия

Мы строили различные партии игр, но при этом совсем не принимали во внимание стремление игроков к победе. Теперь нас будут интересовать лишь такие партии, в которых *оба игрока стремятся к победе*. Будем называть такие партии *разумными*. Игроков, которые стараются победить, а не делают ходы наугад, мы тоже будем называть *разумными*.

Итак, играют двое и каждый из них стремится к победе. Если правила игры не допускают ничьей, то в каждой партии кто-то из игроков обязательно выигрывает. Оказывается, что в каждой игре с полной информацией, правила которой не допускают ничьей, существует *выигрышная стратегия* для одного из игроков.



**Выигрышная стратегия** — это правило, следуя которому один из игроков может выиграть, как бы ни играл его противник.

Используя это правило, можно победить любого соперника — и разумного, и не очень. Ясно, что для каждой игры с определёнными правилами выигрышную стратегию может иметь только один из игроков.

Мы упоминали о том, что шахматы и шашки тоже игры с полной информацией. Значит, и в этих играх есть выигрышная стратегия для одного из игроков. Почему же тогда никто не воспользуется такой стратегией? Поиск выигрышной стратегии в игре с полной информацией часто можно выполнить только при помощи перебора всех возможных партий игры. Для этого строится дерево игры и изучаются вершины построенного дерева. Дерево игры в шахматы огромно: на первом уровне такого дерева 20 вершин, на втором — уже 400 вершин! Построить такое дерево пока не удалось никому, даже при помощи самого мощного компьютера, поэтому и выигрышную стратегию найти тоже пока не удалось. Неизвестно даже, кто из игроков в шахматной игре имеет такую стратегию.

В играх, которые допускают ничью, может существовать *ничейная стратегия* — правило, позволяющее каждому из игроков свести любую партию к ничьей. Ничейная стратегия есть, например, в игре *Крестики-нолики*.

## Выигрышные и проигрышные позиции

Для того чтобы найти выигрышную стратегию, нужно последовательно рассмотреть все возможные позиции игры. Все позиции игры можно перебрать, построив дерево игры. Трудность здесь в том, что для многих игр такое дерево оказывается слишком большим. Но для некоторых игр удаётся построить выигрышную стратегию без перебора всех возможных позиций. Такие игры мы обсудим позже.

Одна из немногих игр, в которых можно перебрать все позиции, не строя дерева игры, — это игра *Камешки*: ведь все возможные позиции этой игры укладываются в начало числового ряда от нуля до начальной позиции.



Рассмотрим игру *Камешки* с начальной позицией 8 камешков, в которой разрешается брать на каждом ходу 1, 3 или 4 камешка. Изучать позиции игры будем с точки зрения разумного игрока, чья очередь делать ход (кто именно из игроков — Первый или Второй — неважно). Разместим все возможные позиции игры на числовой линейке.



Назовём **позицию выигрышной**, если из неё есть ход, который оставит противнику проигрышную позицию.



Назовём **позицию проигрышной**, если любой ход из неё оставляет противнику выигрышную позицию.

Позиция 0 *проигрышная*: партия закончена, игрок, чья очередь была бы делать ход, *уже проиграл*.

Позиции 1, 3 и 4 *выигрышные*: игрок может забрать все камешки и тем самым оставить противнику проигрышную позицию 0.

Позиция 2 *проигрышная*: из этой позиции можно сделать только один ход — взять 1 камешек и тем самым оставить противнику выигрышную позицию 1.

На числовой линейке выигрышные позиции пометим красным, а проигрышные — синим.



Позиция 5 *выигрышная*: взяв 3 камешка, игрок оставляет противнику проигрышную позицию 2. Так же и позиция 6 *выигрышная*: взяв 4 камешка, игрок оставляет противнику проигрышную позицию 2.



Позиция 7 *проигрышная*: все ходы, которые можно сделать из этой позиции, оставляют противнику выигрышную позицию 6, 4 или 3.

Начальная позиция **8** *выигрышная*: взяв **1** камешек, игрок оставляет противнику проигрышную позицию **7**.



Изучая позиции игры от заключительной к начальной, мы последовательно поместили все возможные позиции игры как выигрышные или проигрышные. При этом начальная позиция оказалась выигрышной. Это означает, что в данной игре выигрышную стратегию имеет Первый (тот, кто должен делать ход в начальной позиции).

Пользуясь раскрашенной числовой линейкой, сформулируем выигрышную стратегию для Первого:

1. На первом ходу взять один камешек (при этом Второму достаётся **7** камешков).
2. На третьем ходу взять столько камешков, чтобы оставить на столе только **2** камешка или **0** камешков (сколько именно камешков при этом надо взять, зависит от того, сколько камешков возьмёт Второй на втором ходу).
3. На пятом ходу (если игра не закончилась раньше) забрать оставшийся камешек и выиграть.



Выигрышная стратегия заключается в том, чтобы после каждого хода оставлять противнику проигрышную позицию.

Поэтому выигрышную стратегию в игре имеет тот игрок, который первым сможет занять выигрышную позицию. Поэтому:



Если начальная позиция выигрышная, то выигрышную стратегию имеет Первый, если проигрышная — Второй.

**116**

Найди выигрышную стратегию в игре *Камешки* (начальная позиция **15**, разрешается брать **1**, **3** или **4** камешка). Для этого исследуй все позиции игры, раскрась числовую линейку (можно вырезать заготовку числовой линейки со вкладыша тетради



проектов). Выясни, у кого из игроков есть выигрышная стратегия. Построй последовательность такой партии, в которой этот игрок следует выигрышной стратегии, а его противник на каждом ходу берёт один камешек.

**117**

Найди выигрышную стратегию в игре *Камешки* (начальная позиция **14**, разрешается брать **1**, **3** или **4** камешка). Можешь воспользоваться числовой линейкой, уже раскрашенной в ходе решения задачи **116**. Выясни, у кого из игроков есть выигрышная стратегия в этой игре, сформулируй выигрышную стратегию.

**118**

Найди выигрышную стратегию в игре *Камешки* (начальная позиция **214**, разрешается брать **1** или **2** камешка).



Для решения необязательно раскрашивать числовую линейку от 0 до 214 целиком. Вместо этого можно:

- 1) раскрасить позиции от 0 до 15;
- 2) найти закономерность расположения проигрышных позиций на числовой прямой;
- 3) определить, какой будет начальная позиция, а значит, выяснить, у кого из игроков есть выигрышная стратегия;
- 4) сформулировать выигрышную стратегию, не перечисляя проигрышные позиции, а описывая их.

**119**

*Робот* находится в центре коридора без боковых выходов, идущего в неизвестном направлении: вертикальном (идущего вверх-вниз) или горизонтальном (идущего вправо-влево). Составь алгоритм, выводящий *Робота* из коридора, если известно, что сейчас он находится на расстоянии не больше 15 шагов до каждого из двух выходов коридора.

**120**

Реши задачу.  
Пятеро друзей встретились и обменялись рукопожатиями каждый с каждым. Сколько всего было рукопожатий?



121

Даны правила игры *Сотня*.

### Правила игры *Сотня*

Начальная позиция. Число 0.

Возможные ходы. На каждом ходу игрок прибавляет к имеющемуся числу любое целое число от 1 до 9 включительно.

Как определить победителя. Игра заканчивается, если позиция оказывается равной **100**. При этом выиграл тот, кто прибавил последнее число.

Устройте соревнование с соседом по парте — сыграйте 4 партии в *Сотню*.

Начинайте игру по очереди: пусть один из вас играет Первым в партиях с чётными номерами, а другой — с нечётными. Заполни таблицу соревнования (образец такой таблицы дан в задаче 94, вырежи такую таблицу из вкладыша тетради проектов). За каждую победу игрок получает 1 очко, а за поражение — 0 очков. Сколько раз в вашем соревновании выиграл Второй?

122

Найди выигрышную стратегию в игре *Сотня*.

Для этого:

- 1) начни раскрашивать числовую линейку, начиная с заключительной позиции — от 100 до 78 (можно вырезать заготовку числовой линейки со вкладыша тетради проектов);
- 2) найди закономерность расположения проигрышных позиций на числовой прямой;
- 3) определи, какой будет начальная позиция — выигрышной или проигрышной, а значит, выясни, у кого из игроков есть выигрышная стратегия;
- 4) сформулируй выигрышную стратегию.

123

Построй последовательность бусин  $\Gamma$  длины не меньше пяти, такую, чтобы все следующие утверждения не имели смысла для этой последовательности:

В последовательности  $\Gamma$  седьмая бусина — красная треугольная.

В последовательности  $\Gamma$  следующая бусина после каждой круглой — квадратная.

В последовательности  $\Gamma$  предыдущая бусина перед каждой квадратной — треугольная.

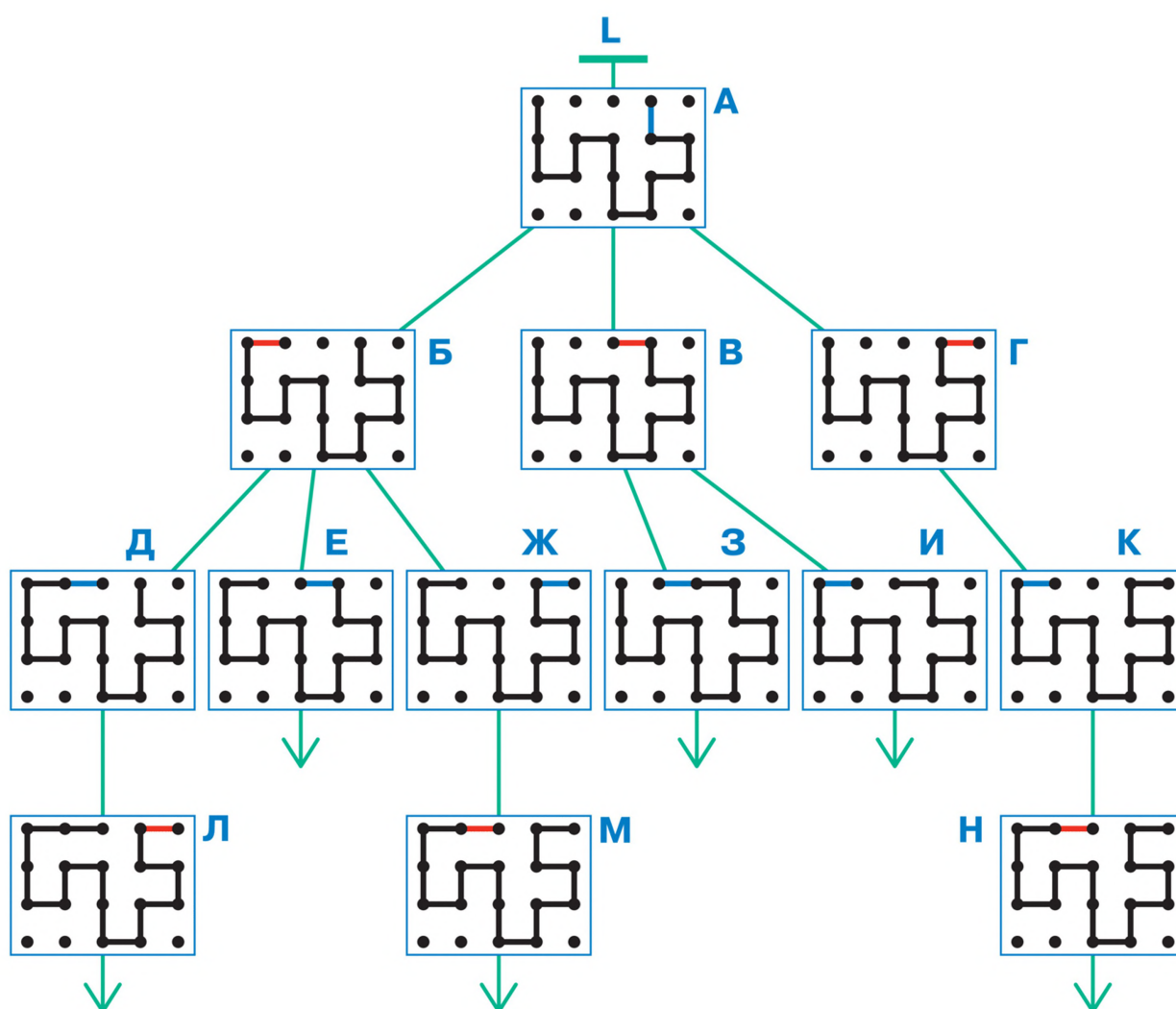


124

Дерево игры *Ползунок* (см. задачу 112) на поле  $5 \times 4$  очень большое, оно не помещается на странице. Дерево  $L$  — это часть полного дерева игры, состоящая из некоторого элемента (т. е. позиции)  $A$  и всех элементов, в которые из  $A$  идут последовательности. Таким образом,  $L$  есть дерево перебора всех возможных окончаний игры из позиции  $A$ .

Определи, какой игрок — Первый или Второй — должен ходить в позиции  $A$ . Затем определи, у кого из игроков есть выигрышная стратегия. Для этого:

- 1) перерисуй дерево  $L$  в тетрадь, заменив позиции их именами;
- 2) исследуй все позиции дерева  $L$  в учебнике и в тетради, имя каждой выигрышной позиции обведи красным, а проигрышной — синим;
- 3) определи, у кого из игроков есть выигрышная стратегия, и запиши эту стратегию в виде последовательности позиций.



125

Даны правила игры *Пешка*.

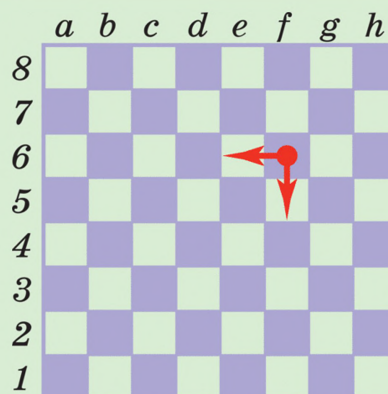


### Правила игры *Пешка*

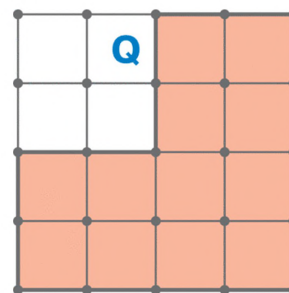
Начальная позиция. Игра ведётся на шахматной доске, пешка стоит на одном из полей (на каком именно поле — устанавливается дополнительными правилами).

Возможные ходы. На каждом ходу игрок передвигает пешку на одно поле влево или на одно поле вниз.

Как определить победителя. Игра заканчивается, если пешка оказывается в левом нижнем углу доски — на поле *a1*. Выигрывает тот игрок, который сделал последний ход.



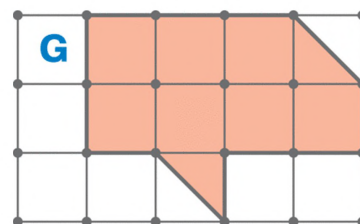
Исследуй игру *Пешка*. Попытайся объяснить, почему в этой игре выигрышная стратегия не нужна: победа не зависит от того, насколько умело играют игроки. При каких начальных позициях в игре *Пешка* выигрывает Первый и при каких — Второй?



126

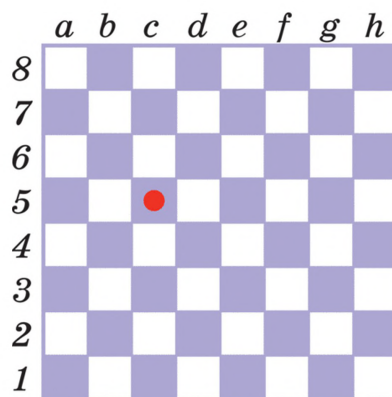
а) Нарисуй, как разрезать многоугольник Q, чтобы получилось четыре одинаковых многоугольника.

б) Нарисуй, как разрезать многоугольник G, чтобы получилось два одинаковых многоугольника.



127

За один ход фишка может сдвинуться на одно поле влево или на одно поле вверх. Сколькими различными путями фишка может пройти по шахматной доске из поля *c5* на поле *a8*, если будет двигаться только влево или вверх?





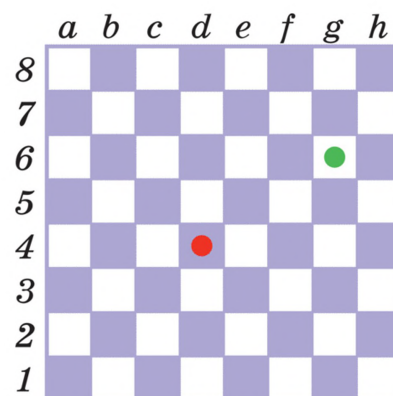
# Выигрышные стратегии

Продолжим изучение игр и поиск выигрышных стратегий путём исследования всех позиций игры. Ты знаешь, что у большинства игр деревья игры слишком велики, чтобы мы могли исследовать все позиции такой игры на дереве. Если каждую позицию можно представить в виде числа (как в играх *Камешки* и *Сотня*), то все позиции такой игры можно разместить на числовой линейке и исследовать по порядку. Это возможно только тогда, когда разных позиций-чисел не слишком много.

Рассмотрим другие примеры игр, все позиции которых мы можем удобно расположить.

Таковыми играми, например, являются игры на шахматной доске, в которых игра ведётся только одной фигурой. В такой игре каждая позиция соответствует тому полю шахматной доски, на котором находится фигура (*полем* в шахматах называется клетка шахматной доски). Исследовав все поля шахматной доски, мы исследуем *все возможные позиции* игры.

Каждое поле шахматной доски имеет имя, состоящее из латинской буквы и цифры. Например, поле с красной точкой имеет имя *d4*, а поле с зелёной точкой — имя *g6*.

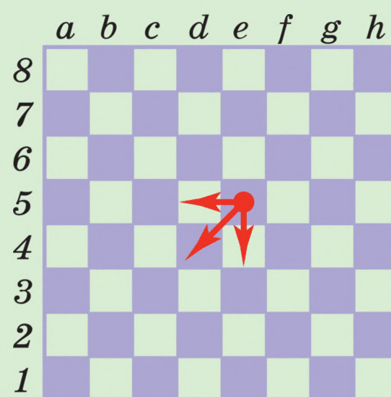


## Правила игры *Король*

Начальная позиция. Игра ведётся на шахматной доске, король стоит на одном из полей (на каком именно, устанавливается дополнительными правилами).

Возможные ходы. На каждом ходу игрок передвигает короля на одно поле влево, или на одно поле вниз, или на одно поле влево-вниз по диагонали.

Как определить победителя. Игра заканчивается, если король оказывается в левом нижнем углу доски — на поле *a1*. Выигрывает игрок, который сделал последний ход.



Исследуем все позиции игры *Король* с начальной позицией *d4*. Возьмём рисунок шахматной доски, не раскрашенный в шахматном порядке: ведь шахматная раскраска в этой игре роли не играет, а нам при раскрашивании полей она будет мешать. Выигрышные позиции будем раскрашивать красным цветом, проигрышные — синим.

Начнём с заключительной позиции — поля *a1*. Эта позиция проигрышная. Все позиции, из которых за один ход можно попасть на поле *a1*, — поля *a2*, *b2*, *b1* — выигрышные позиции. Далее выбираем для исследования только те поля, все ходы из которых ведут на уже раскрашенные поля.

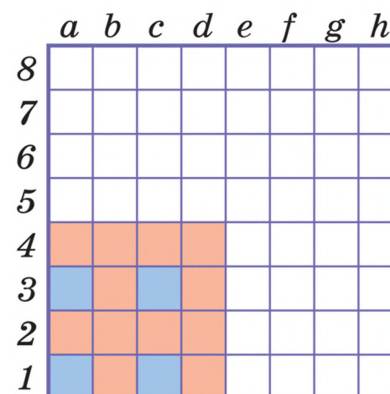
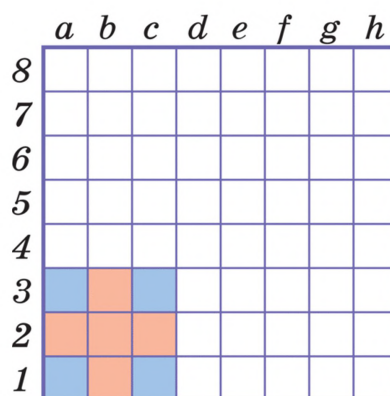
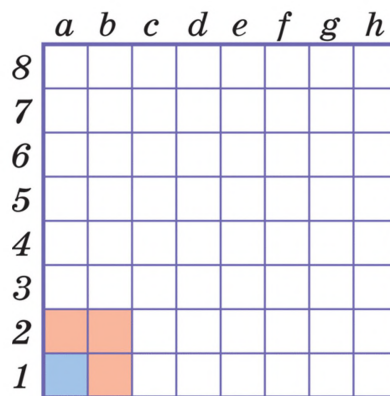
С поля *a3* можно пойти только на поле *a2* (выигрышное) — значит, позиция *a3* проигрышная. Аналогично этому позиция *c1* проигрышная. Из позиции *b3* можно перейти в проигрышную позицию *a3* — значит, позиция *b3* выигрышная. Аналогично этому выигрышной является позиция *c2*. Все ходы из позиции *c3* ведут в выигрышные позиции, поэтому позиция *c3* проигрышная.

Мы будем раскрашивать поля, пока не раскрасим все поля квадрата  $4 \times 4$ .

Остальные поля доски раскрашивать не надо, потому что они не являются позициями игры *Король* с начальной позицией *d4*.

Начальная позиция (поле *d4*) выигрышная; значит, выигрышная стратегия есть у Первого, и заключается она в следующем:

1. Пойти на поле *c3*;
2. В зависимости от хода Второго пойти на поле *c1* либо на поле *a3*;
3. Какой бы ход ни сделал Второй, пойти на поле *a1*.

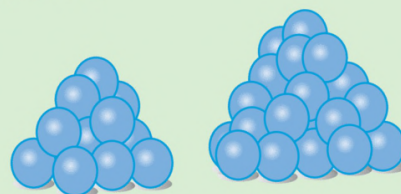




Есть и другие игры, для которых все позиции можно расположить в таблице, похожей на изображение шахматной доски. Такова, например, игра *Две кучи камешков*.

### Правила игры *Две кучи камешков*

Начальная позиция. Две кучи камешков (количество камешков в каждой куче устанавливается дополнительными правилами).



Возможные ходы. На каждом ходу игрок может взять либо один камешек из одной из куч, либо по одному камешку из обеих куч одновременно.

Как определить победителя. Игра заканчивается, если все камешки закончились. Выигрывает тот игрок, который забрал последний камешек.

Каждая позиция этой игры — это две кучи камешков; её можно представить в виде пары чисел, где первое число — количество камешков в первой куче, а второе — во второй. Например, пара (5; 2) — позиция игры, где в первой куче 5 камешков, а во второй — 2.

Все возможные позиции такой игры можно расположить в таблице, столбцы и строки которой занумерованы числами 0, 1, 2 и т. д. Каждая клетка таблицы соответствует некоторой позиции: номер столбца, в котором находится данная клетка, — это число камешков в первой куче, а номер строки — число камешков во второй.

На рисунке справа показана таблица для игры *Две кучи камешков*, где в начальной позиции в первой куче 6 камешков, а во второй — 7. Зелёной точкой помечена клетка, соответствующая позиции (5; 2).

При помощи такой таблицы все позиции данной игры можно исследовать так же, как мы исследовали позиции игры *Коль*. Это ты сделаешь, когда будешь решать задачи.

	0	1	2	3	4	5	6
0							
1							
2						●	
3							
4							
5							
6							
7							

128



Исследуй все позиции игры *Король* с начальной позицией *h8*:

1) раскрась шахматную доску, начиная с заключительной позиции — с клетки *a1*;

2) определи, выигрышной или проигрышной будет начальная позиция, а значит, выясни, у кого из игроков есть выигрышная стратегия.

129



Даны правила игры *Ладья*.

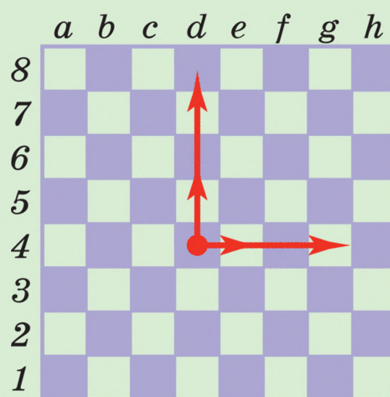


### Правила игры *Ладья*

Начальная позиция. Игра ведётся на шахматной доске, ладья стоит на одном из полей (на каком именно — устанавливается дополнительными правилами).

Возможные ходы. На каждом ходу игрок передвигает ладью на сколько угодно полей вправо или на сколько угодно полей вверх.

Как определить победителя. Игра заканчивается, если ладья оказывается в правом верхнем углу доски — на поле *h8*. Выигрывает игрок, который сделал последний ход.



Сыграй с соседом по парте две партии в игру *Ладья* с начальной позицией *a1* и две партии с начальной позицией *a2*. Нарисуй на шахматной доске путь, который прошла ладья в ходе одной из сыгранных вами партий.

130

Определи, при каких начальных позициях в игре *Ладья* выигрышная стратегия есть у Первого и при каких — у Второго:

1) раскрась шахматную доску, начиная с заключительной позиции — поля *h8*;

2) постарайся коротко описать, при каких начальных позициях выигрышную стратегию имеет Первый и при каких — Второй;

3) сформулируй выигрышную стратегию для Первого в игре с начальной позицией на поле *a2*;

4) сформулируй выигрышную стратегию для Второго в игре с начальной позицией на поле *a1*.



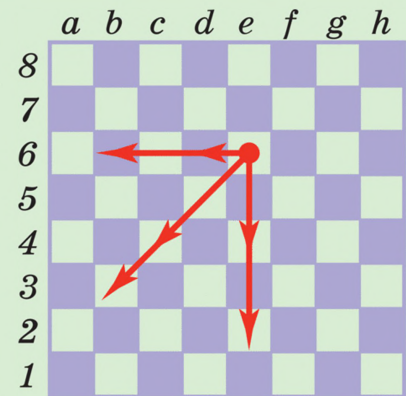


### Правила игры Ферзь

Начальная позиция. Игра ведётся на шахматной доске, ферзь стоит на одном из полей (на каком именно — устанавливается дополнительными правилами).

Возможные ходы. На каждом ходу игрок передвигает ферзя на сколько угодно полей: влево, вниз или по диагонали влево-вниз.

Как определить победителя. Игра заканчивается, если ферзь оказывается в левом нижнем углу доски — на поле *a1*. Выигрывает игрок, который сделал последний ход.

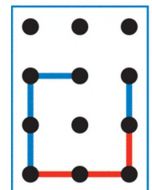


Найди выигрышную стратегию в игре Ферзь с начальной позицией *g8* и в той же игре с начальной позицией *h5*:

- 1) раскрась все поля шахматной доски, начиная с заключительной позиции — поля *a1*;
- 2) для каждой из двух начальных позиций (*g8* и *h5*) определи, какой она будет — выигрышной или проигрышной, а значит, у кого из игроков есть в игре с этой начальной позицией выигрышная стратегия;
- 3) сформулируй выигрышную стратегию для каждой из этих двух начальных позиций.

Выпиши имена шести полей, с начальными позициями в которых в игре Ферзь выигрышная стратегия есть у Первого. Выпиши имена ещё шести полей, с начальными позициями в которых в игре Ферзь выигрышная стратегия есть у Второго.

Построй часть дерева игры Ползунок с данной позицией в элементе первого уровня. По образцу, приведённому на странице 78, дай всем элементам полученного дерева имена и найди выигрышную стратегию окончания игры из этой позиции.



Для этого проведи исследование, как в задаче 124.

133

Найди выигрышную стратегию в игре *Две кучи камешков* с начальной позицией (4; 5): раскрась поле, начиная с заключительной позиции — клетки (0; 0), и определи, какой будет начальная позиция — выигрышной или проигрышной, а значит, у кого из игроков есть выигрышная стратегия. Запиши последовательность позиций какой-нибудь партии, в которой один из игроков использует выигрышную стратегию, а другой на первом своём ходу берёт по одному камешку из каждой кучи, а на следующем берёт один камешек из одной из куч (позиции обозначай парами чисел).

	0	1	2	3	4
0					
1					
2					
3					
4					
5					

134

Построй два разных множества, для каждого из которых истинны все следующие утверждения:

Все элементы этого множества — двузначные нечётные числа.

Сумма цифр каждого числа из этого множества равна 10.

Самое большое число из этого множества на 1 меньше суммы всех остальных чисел из этого множества.

135

Составь алгоритм со следующим заголовком:

```

алг переход в противоположный угол
  дано | Робот стоит в каком-то углу поля
        | размером 10 × 14 клеток, на поле
        | стен нет
  надо | Робот перешёл в противоположный
        | угол
  
```

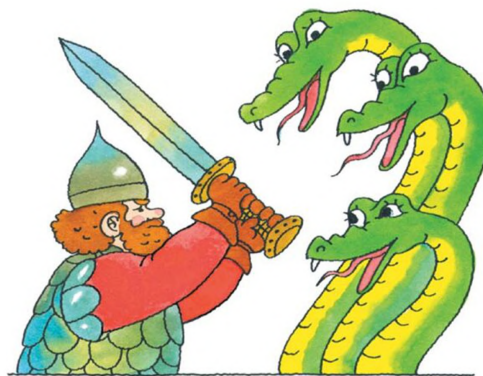
136

*Робот* находится внутри тупика: горизонтального коридора без боковых выходов, закрытого с одного из концов (правого или левого — неизвестно). Составь алгоритм, выводящий *Робота* из этого коридора, если известно, что в начальном состоянии он находится на расстоянии 10 шагов до выхода из коридора и в 10 шагах до закрытого его конца.



137

Реши задачу, используя поиск выигрышной стратегии в игре. Алёша Попович и Добрыня Никитич воюют с девятиглавым змеем. По очереди богатыри ходят к его пещере и отрубают 1, 2 или 3 головы. Как начавшему бой Алёше обрести славу победителя змея (отрубить последнюю голову)?



138

Найди выигрышную стратегию в игре *Камешки* (начальная позиция **10**, разрешается брать **1**, **2** или **3** камешка).



Для этого исследуй все позиции игры, раскрась числовую линейку. Выясни, у кого из игроков есть выигрышная стратегия. Построй последовательность позиций такой партии, в которой игрок следует выигрышной стратегии, а его противник на каждом ходу берёт **2** камешка.

139

Прочитай описание игры *Назови 26*.

Играют двое.

Первый игрок называет любое натуральное число, не превосходящее 4, т. е. одно из чисел 1, 2, 3, 4.

Второй игрок прибавляет к названному числу своё число, также не превосходящее 4.

К этой сумме первый прибавляет какое-либо натуральное число, не превосходящее 4, и сообщает сумму и т. д.

Выигрывает тот, кто первым достигнет числа 26.

Сформулируй правила этой игры как игры с полной информацией. Найди выигрышную стратегию в этой игре.

140

Нарисуй в тетради по клеткам два разных прямоугольных треугольника, площадь каждого из которых равна 12 ед. кв.

## Цикл «пока»

Решим задачу.

**Задача.** На поле где-то ниже *Робота* на неизвестном расстоянии есть стена. Нужно, чтобы *Робот* подошёл вплотную к этой стене.

Если мы командуем *Роботом* вручную, без компьютера, то задачу можно решить так: дадим *Роботу* запрос снизу свободно. Если *Робот* ответит «нет», значит, он уже у стены и задача решена. Если же *Робот* ответит «да», то можно скомандовать вниз и опять дать запрос снизу свободно. Если *Робот* ответит «да» — опять скомандовать вниз, дать запрос снизу свободно, и так поступать дальше, пока *Робот* не ответит «нет». Заранее неизвестно, сколько раз мы дадим команду вниз — 0, 3, 8 или 2014 раз, это зависит от начального расположения *Робота* относительно стены. Например, если *Робот* уже стоял рядом со стеной, то команда вниз не подавалась бы ни разу. Если *Робот* стоял в 101-й клетке от стены, то команда вниз подавалась бы 100 раз.

Наши действия при решении этой задачи можно описать так: пока на запрос снизу свободно *Робот* отвечает «да», надо командовать вниз и повторять запрос. Такое описание пригодно при *любом* расстоянии от *Робота* до стены.

Наша цель, однако, не ручное управление *Роботом*, а составление программы (алгоритма) для компьютера. Поэтому приведённую выше последовательность действий надо описать на алгоритмическом языке. Алгоритм должен быть универсальным, т. е. не должен зависеть от расстояния между *Роботом* и стеной. Для этого в алгоритмическом языке есть специальная *составная команда* — **цикл «пока»**. В общем виде **цикл «пока»** записывается так:

```
нц пока <условие>  
    <последовательность команд цикла>  
кц
```

Слова **нц** и **кц** имеют тот же смысл, что и в **цикле «N раз»**, — они отмечают начало и конец цикла. При выполнении **цикла «пока»** компьютер повторяет следующие действия:

а) проверяет *условие*, записанное после служебного слова **пока**;



б) если условие не соблюдается (*Робот* ответил «нет»), то выполнение цикла завершается и компьютер начинает выполнять команды, записанные после **кц**. Если же условие соблюдается (*Робот* ответил «да»), то компьютер выполняет последовательность команд цикла, потом снова проверяет условие и т. д.

Вот алгоритм вниз до стены для решения задачи, в нём использована составная команда **цикл «пока»**:

```

алг вниз до стены
  дано |
  надо | Робот идёт вниз ,
        | пока не дойдёт
        | до стены

  нач
    нц пока снизу свободно
      вниз
    кц
  кон

```

Аналогично можно составить алгоритмы влево до стены, вправо до стены, вверх до стены, которыми мы далее часто будем пользоваться.

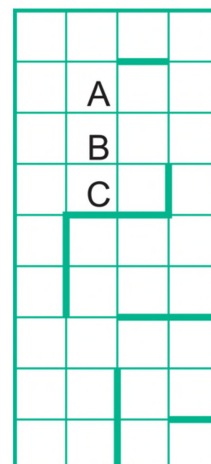
Как *Робот* выполняет такой алгоритм?

Например, пусть *Робот* стоит в клетке А. Тогда при выполнении алгоритма вниз до стены диалог между *Роботом* и компьютером будет таким:

```

компьютер: снизу свободно?
Робот: да
компьютер: вниз
Робот: <смещается вниз в клетку В>
компьютер: снизу свободно?
Робот: да
компьютер: вниз
Робот: <смещается вниз в клетку С>
компьютер: снизу свободно?
Робот: нет

```



Поскольку *Робот* ответил «нет», т. е. записанное после **пока** условие не соблюдается, то на этом выполнение **цикла «пока»** и алгоритма вниз до стены заканчивается.

## Свойства цикла «пока»

Свойство 1. *Последовательность команд цикла может не выполняться ни разу.*

Если условие в **цикле «пока»** не соблюдается с самого начала, то последовательность команд цикла не выполняется ни разу. Например, если в алгоритме вниз до стены *Робот* на первый же запрос снизу свободно ответит «нет» (т. е. если по нижней границе клетки, в которой находится *Робот*, с самого начала уже стоит стена), то компьютер не вызовет команду вниз ни разу. Важно понимать, что ситуация, когда цикл ни разу не выполняется, не является *отказом*. Это нормальный вариант выполнения алгоритма.

Свойство 2. *Выполнение цикла может не завершиться.*

Выполнение **цикла «пока»** может не завершиться, если условие всё время будет соблюдаться. Такая ситуация обычно возникает из-за ошибок в составлении алгоритма. Она называется *заикливанием*.

Например, рассмотрим такой фрагмент алгоритма:

```
нц пока снизу свободно  
    вниз  
    вверх  
кц
```

Если снизу от *Робота* нет стены, он будет *бесконечно* топтаться на месте, совершая шаги вверх и вниз. Исполнение алгоритма никогда не закончится.

Свойство 3. *Условие цикла не проверяется в процессе выполнения последовательности команд цикла.*

Условие в **цикле «пока»** проверяется только перед выполнением последовательности команд цикла, но не проверяется в процессе выполнения этих команд.

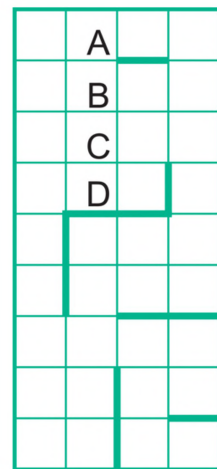


**Пример.** Пусть *Робот* находится в клетке А (см. рисунок) и компьютер выполняет такой цикл:

```
нц пока снизу свободно
  вниз
  вниз
кц
```

Тогда диалог между компьютером и *Роботом* будет таким:

```
компьютер: снизу свободно?
Робот: да
компьютер: вниз
Робот: <смещается вниз в клетку В>
компьютер: вниз
Робот: <смещается вниз в клетку С>
компьютер: снизу свободно?
Робот: да
компьютер: вниз
Робот: <смещается вниз в клетку D>
компьютер: вниз
Робот: ОТКАЗ
```



В процессе выполнения последовательности команд цикла компьютер не проверяет, истинно ли условие выполнения цикла. Если после выполнения какой-то команды внутри цикла «пока» условие перестанет соблюдаться, то компьютер всё равно будет выполнять последовательность команд цикла до конца.

Свойство 4. *Перед каждым выполнением последовательности команд цикла условие обязательно выполняется.*

Это очевидное свойство: если компьютер приступает к выполнению последовательности команд цикла, значит, условие выполнено.

Свойство 5. *Сразу после окончания цикла условие не выполняется.*

Это свойство тоже очевидно: цикл заканчивается в тот момент, когда при очередной проверке условие оказалось невыполненным.

## Составление алгоритма с циклом «пока»

Цикл «пока» используется всякий раз, когда число повторений каких-то действий заранее неизвестно. Составление таких циклов — трудная задача.

Чтобы не запутаться и что-нибудь не забыть, лучше всего строить цикл «пока» по частям:

1. Понять, когда цикл должен закончиться, т. е. сформулировать условие выполнения цикла. Иногда это удобно делать от противного — записать противоположное условие.
2. Описать, что происходит при однократном выполнении цикла (принято говорить «за один шаг цикла»), т. е. *выполнить один раз последовательность команд цикла*.
3. Проверить, что цикл рано или поздно закончится, а не будет повторяться вечно.

Циклы «N раз» и «пока» оформляются в алгоритмическом языке почти одинаково — обе эти команды задают повторяющуюся последовательность команд. Однако у этих двух циклов есть одно существенное различие. Начиная выполнять цикл «N раз», компьютер *знает*, сколько раз придётся повторить последовательность команд цикла. При исполнении цикла «пока» это не так: компьютер каждый раз проверяет условие цикла и *не может заранее определить*, через сколько повторений выполнение закончится. Определить количество повторений цикла «пока» можно только после того, как цикл завершён.

Отсюда ясно, в каких случаях какой цикл следует использовать. Если к моменту начала цикла количество повторений известно, можно воспользоваться циклом «N раз». Если же количество повторений заранее определить нельзя, необходим цикл «пока».



**141**

Составь диалог компьютера и *Робота* при выполнении каждого цикла в ситуации, когда *Робот* стоит: а) в закрашенной клетке; б) в незакрашенной клетке. Если выполнение алгоритма приведёт к заикливлению — прерви описание диалога и напиши о заикливлении.

```
нц пока клетка чистая
    закрасить
кц
```

```
нц пока клетка закрашена
    закрасить
кц
```

**142**

Составь алгоритмы влево до стены, вправо до стены, вверх до стены, о которых идёт речь на с. 88. Запиши условия **дано** и **надо**. Проверь, что алгоритмы составлены правильно — что не возникнет отказа и что цикл не будет повторяться вечно.

**143**

*Робот* находится у левой стены прямоугольного поля, огороженного со всех сторон стенами. Внутри поля стен нет, размеры поля неизвестны. Надо составить такой алгоритм, при выполнении которого *Робот* закрасит горизонтальный ряд клеток от исходного положения до правой стены и вернётся в исходное положение: сначала *Робота* надо довести до правой стены, по дороге закрашивая клетки, а затем вернуть в исходное положение и закрасить эту исходную клетку. Вот этот алгоритм:

```
алг закрасить ряд вправо и вернуться
  дано | Робот стоит у левой стены поля
  надо | закрашен весь горизонтальный ряд,
        | в котором стоит Робот,
        | Робот в исходном положении
нач
  вправо до стены с закрашиванием
  влево до стены
  закрасить
кон
```

Вспомогательный алгоритм влево до стены ты построил в ходе решения задачи 142. Построй вспомогательный алгоритм вправо до стены с закрашиванием. Запиши условия **дано** и **надо**. Проверь, что алгоритм составлен правильно — что не возникнет отказа и что цикл не будет повторяться вечно.

144

Составь алгоритм с таким заголовком (см. ниже).

**алг** закрасить прямоугольник

**дано** | Робот стоит в левом нижнем углу

| поля без внутренних стен

**надо** | весь прямоугольник закрашен, Робот

| в исходном положении

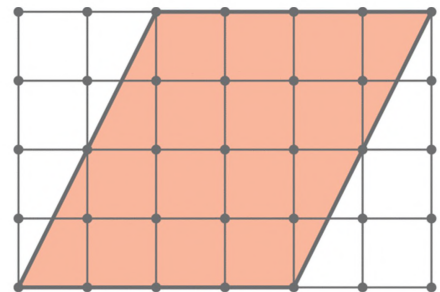
Используй в качестве вспомогательного алгоритм закрасить ряд вправо и вернуться, составленный при решении задачи 143.

145

Реши задачу 144, считая, что о начальном положении *Робота* на поле ничего не известно.

146

Нарисуй такой же четырёхугольник по клеткам в тетради. Потом нарисуй, как разрезать его на части, чтобы из них можно было собрать квадрат на сетке. Нарисуй такой квадрат и покажи штриховыми линиями, из каких частей он составлен.



147

На поле *Робота* ровно одна стена. Это горизонтальная стена, которая не примыкает к границам поля. Робот находится в одной из клеток, прилегающих к стене сверху. Точные размеры поля и стены и точное расположение *Робота* неизвестны. Составь алгоритм, при выполнении которого *Робот*:

- а) окажется в одной из клеток, прилегающих к стене снизу;
- б) закрасит все клетки, прилегающие к стене снизу;
- в) закрасит все прилегающие к стене клетки.



148

Робот стоит в левом конце горизонтального коридора, нижняя стена которого сплошная, а в верхней имеется несколько выходов. Составь алгоритм, который заставляет Робота переместиться из клетки А в клетку В (см. рисунок, сколько клеток между клетками А и В, неизвестно) и закрасить все те клетки коридора, которые расположены напротив боковых выходов.



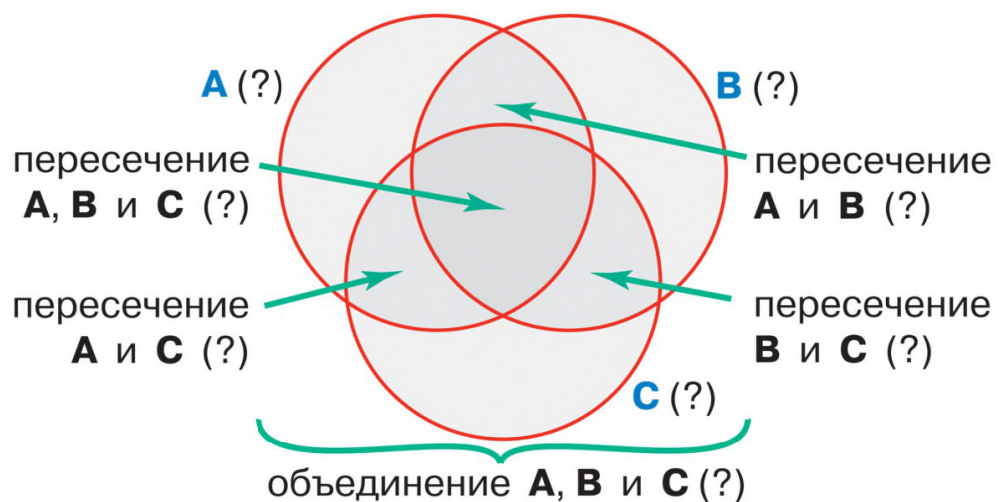
149

Реши задачу.

В нашем классе из 31 учащегося по итогам первой четверти неуспевающих нет (ни у кого нет ни одной четвертной двойки). При этом у нас имеется 3 круглых отличника и 3 круглых троечника (тех, у кого тройки в четверти по всем предметам). Пятеро учеников закончили четверть только на четвёрки и пятёрки. Известно также, что пятёрки имеют среди четвертных оценок 18 учеников и 24 ученика имеют среди четвертных оценок четвёрки. Сколько учеников нашего класса имеют в четверти и тройки, и четвёрки, и пятёрки? Сколько учеников нашего класса не получили в четверти ни одной четвёрки?



Для решения задачи нарисуй схему с множествами:



150

Нарисуй в тетради по клеткам:

а) прямоугольник, не являющийся квадратом, площадь которого равна 64 ед. кв.;

б) прямоугольный треугольник, площадь которого равна 17 ед. кв.;

в) четырёхугольник, площадь которого равна 13 ед. кв.;

г) два разных прямоугольных треугольника, площадь каждого из которых равна  $10\frac{1}{2}$  ед. кв.;

д) два разных четырёхугольника, не являющихся прямоугольниками, площадь каждого из которых равна 21 ед. кв.

151

К полю для игры *Крестики-нолики* добав-

лена одна клетка. Правила игры при этом

остались прежними: Первый ставит крес-

тики, Второй — нолики. Выигрывает тот,

кто первым построит ряд из трёх своих

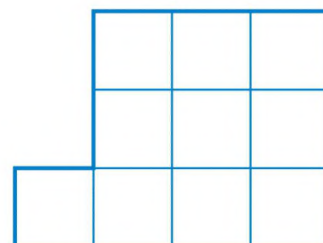
значков по горизонтали, вертикали или

диагонали. В отличие от игры на обыч-

ном поле, в игре *Крестики-нолики* на таком поле Первый име-

ет выигрышную стратегию. Куда должен поставить крестик

Первый на первом ходу, чтобы наверняка выиграть?



152

Вот правила игры *7 камней*:

### **Правила игры *7 камней***

Начальная позиция. Куча из 7 камешков.

Возможные ходы. На каждом ходу игрок разделяет любую из имеющихся куч на две (необязательно пополам).

Как определить победителя. Игра заканчивается, если очередной ход сделать невозможно. Выигрывает тот, кто сделал последний ход.

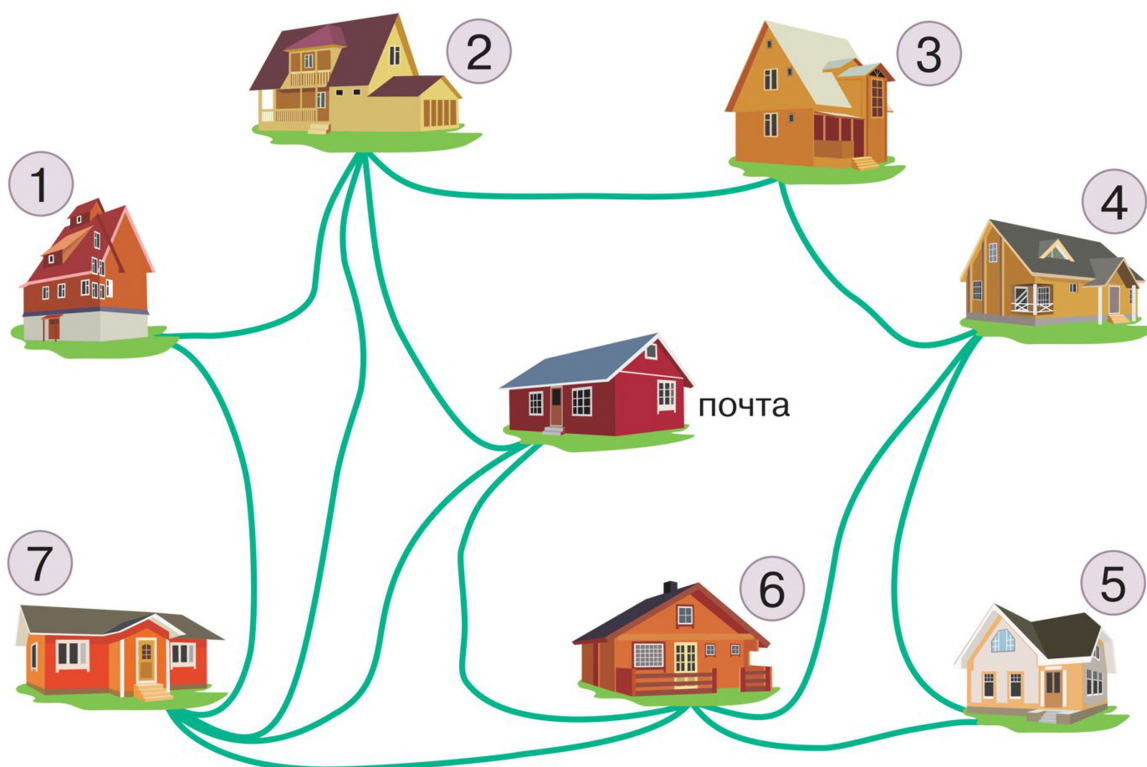
Исследуй игру *7 камней*. Попытайся объяснить, почему в этой игре выигрышная стратегия не нужна, — победа не зависит от того, насколько умело играют игроки. Кто выигрывает в игре *7 камней*?



153

Реши задачу.

Почтальон Печкин, выйдя из почтового отделения, разнёс почту в каждый дом деревни, после чего зашёл с посылкой к дяде Фёдору (зайдя сначала за ней на почту), а потом вернулся домой. На рисунке показаны все тропинки, по которым проходил Печкин, причём, как оказалось, ни по одной из них он не проходил дважды. Каков мог быть маршрут почтальона Печкина? В каком доме живёт дядя Фёдор?



## Равновесные выигрышные стратегии

До сих пор, для того чтобы построить выигрышную стратегию, мы исследовали все позиции игры и выделяли выигрышные и проигрышные позиции. Но для многих игр существуют стратегии, построение которых не требует исследования всех позиций. Таковы, например, *равновесные* стратегии, которыми мы сейчас займёмся.

Рассмотрим игру *Монеты на весах*.

### Правила игры *Монеты на весах*

Начальная позиция. На каждой чашке весов лежат монеты (сколько именно монет лежит на каждой чашке, устанавливается дополнительными правилами). Все монеты одинаковые.

Возможные ходы. На каждом ходу игрок может взять сколько угодно монет с одной чашки.

Как определить победителя. Игра заканчивается, если все монеты закончились. Выигрывает игрок, который забрал последнюю монету.

Пусть в начальной позиции на каждой чашке лежит по 20 монет. При такой начальной позиции весы находятся в *равновесии* — ведь монет на чашках поровну и все монеты одинаковые. Первый любым своим ходом это равновесие *нарушит* — он возьмёт монеты только с одной чашки. А Второй может *восстановить* равновесие — взять столько же монет с другой чашки. Пусть и дальше Второй каждым своим ходом восстанавливает равновесие. Выигрышна ли такая стратегия для Второго?

Восстанавливая равновесие на каждом ходу, Второй добивается того, что все равновесные позиции достаются Первому. Но и заключительная позиция равновесная (на обеих чашках ничего нет), так что и эта позиция в какой-то момент игры достанется Первому. При этом Первый уже не сможет сделать очередной ход и проиграет. Таким образом, описанная стратегия является выигрышной для Второго.

Итак, для предложенной равновесной стратегии мы проверили два условия:

1. Второй всегда сможет сделать свой ход.
2. Заключительная позиция обязательно достанется Первому (и поэтому он обязательно проиграет).

Оба эти условия выполнены, значит, наша стратегия выигрышная.



Стратегию, при которой игрок на каждом своём ходу восстанавливает равновесие (т. е. делает позицию равновесной), мы будем называть **равновесной стратегией**.

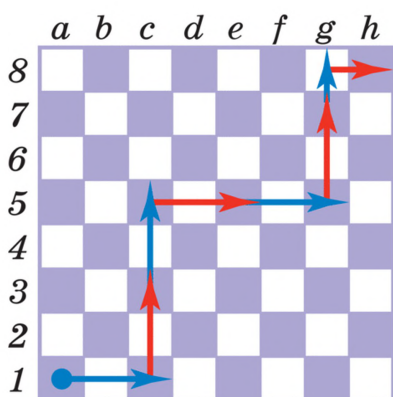


В игре *Монеты на весах*, где начальная позиция неравновесная (например, на одной чашке лежит 15 монет, а на другой — 29), равновесную стратегию имеет Первый: на первом ходу он должен сделать позицию равновесной (взять 14 монет из 29), а дальше повторять ходы Второго, восстанавливая равновесие. Эта стратегия выигрышна для Первого: он всегда сможет сделать ход, а заключительная позиция обязательно достанется Второму.

Равновесную стратегию можно строить не только в играх со взвешиванием, но и во многих других играх. Обычно это игры, в которых проигравшим считается тот, кто не может сделать очередной ход. Для построения равновесной стратегии в такой игре нужно сначала сообразить, какие позиции считать равновесными. При этом важно, чтобы и заключительная позиция тоже оказалась равновесной.

Равновесную стратегию можно построить, например, для игры *Ладья*.

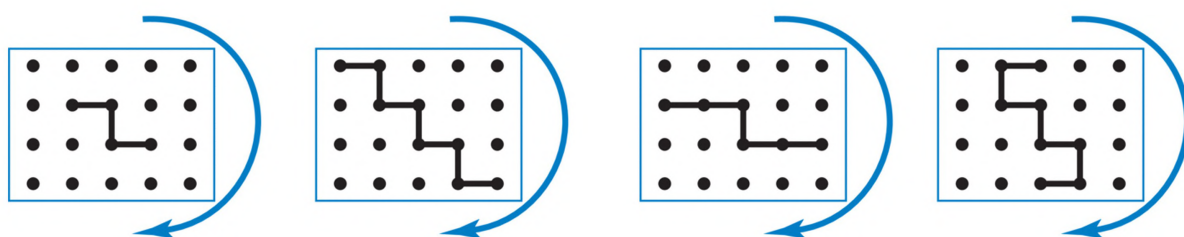
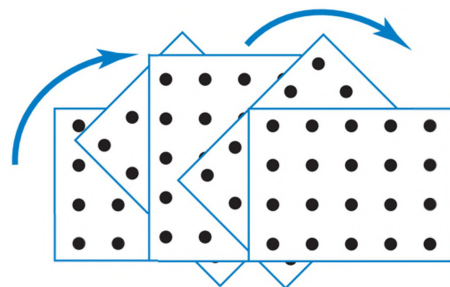
Действительно, можно считать равновесными такие позиции этой игры, в которых ладья стоит на чёрной диагонали шахматной доски. Тогда в игре с начальной позицией *a1* равновесная стратегия Второго состоит в том, чтобы каждый раз после хода Первого возвращать ладью на эту диагональ (и тем самым делать позицию снова равновесной). Такая стратегия оказывается выигрышной для Второго. Следуя этой стратегии, Второй каждый раз передвигает ладью на столько же полей, на сколько её передвинул Первый на предыдущем ходу; но если Первый сделал ход по вертикали, то Второй ходит по горизонтали, и наоборот. Второй всегда сможет сделать такой ход, и заключительная позиция непременно достанется его противнику.



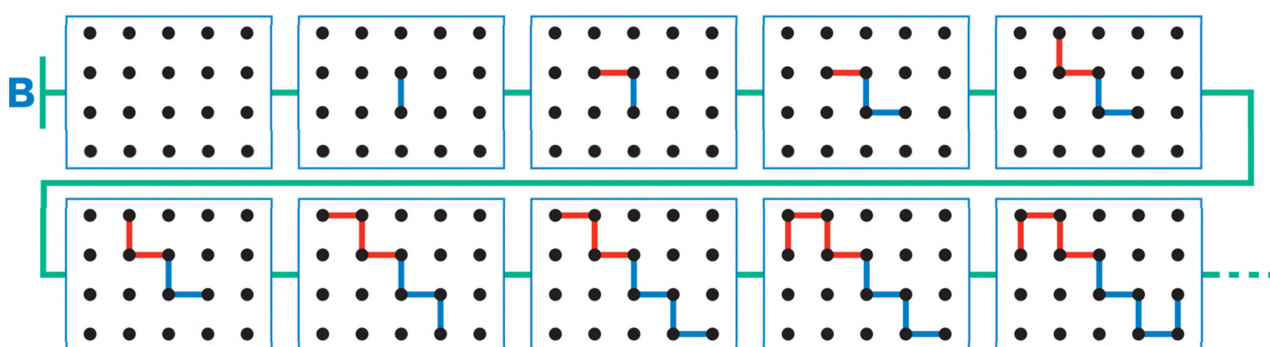
На рисунке показан пример партии, в которой Второй следует этой стратегии (на рисунке показана схема пути ладьи в этой партии).

Заметим, что эта равновесная стратегия ничем не отличается от той стратегии, которую мы построили, изучив все позиции игры: мы просто получили один и тот же результат разными способами.

Приведём ещё один пример игры, для которой можно построить равновесную стратегию. Это знакомая тебе игра *Ползунок* на поле  $5 \times 4$ . Какие позиции в этой игре мы будем считать равновесными? Начальную позицию этой игры можно повернуть «вверх ногами» — от этого рисунок поля не изменится (см. рисунок). Будем считать равновесными и другие позиции, которые не изменяются при таком повороте. Вот примеры таких позиций:



Пусть теперь Первый следует такой равновесной стратегии: на первом ходу он проводит вертикальный отрезок в центре поля (при этом позиция остаётся равновесной). А потом после каждого хода Второго Первый восстанавливает равновесие: если Второй сделал ход на одном конце ползунка (ломаной), то Первый проводит отрезок на другом конце и в обратном направлении. Вот пример начала партии, в которой Первый следует такой стратегии:



При такой стратегии если Второму удалось сделать ход, то и Первому это удастся — он всегда сможет провести отрезок на другой стороне поля и восстановить равновесие. Если же Второй сделать ход не может, то он уже проиграл. Следовательно, описанная равновесная стратегия выигрышна для Первого.



В задачах ты встретишься и с другими играми, в которых можно построить выигрышную равновесную стратегию. Некоторые игры будут похожи на те, которые мы разобрали, а другие будут непохожи. В каждой такой игре нужно будет выделить равновесные позиции и описать равновесную стратегию для одного из игроков. После этого останется только убедиться, что построенная равновесная стратегия является выигрышной — как бы ни играл противник, игрок, следующий этой стратегии, непременно выиграет.

154

Даны правила игры *Шары и ящики*.

### Правила игры *Шары и ящики*

Начальная позиция. Два открытых ящика, в каждом лежат шары. Сколько шаров в каждом ящике, определяется дополнительными правилами.

Возможные ходы. На каждом ходу игрок забирает из одного ящика сколько угодно шаров.

Как определить победителя. Игра заканчивается, если очередной ход сделать невозможно — шары закончились. Выигрывает тот, кто сделал последний ход.

Исследуй игру *Шары и ящики* для различных начальных позиций. Кто из игроков обладает выигрышной стратегией? Опиши эту стратегию.

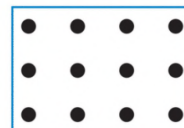


Нужно рассмотреть два варианта начальных позиций:

- 1) если шаров в ящиках поровну;
- 2) если шаров в ящиках не поровну.

155

В игре *Ползунок* на поле  $4 \times 3$  существует равновесная выигрышная стратегия для Первого, подобная той, которая описана на с. 99. Построй последовательность позиций какой-либо партии этой игры, в которой Первый следует равновесной выигрышной стратегии.



156

Сформулируй равновесную выигрышную стратегию для Первого в игре *Ладья* с начальной позицией *a3*. Приведи пример партии этой игры, в которой Первый следует твоей стратегии, — нарисуй путь ладьи в такой партии.

157

Классифицируй числа множества  $P$  по остаткам от деления на 4: в одну группу помести все числа множества  $P$ , которые делятся на 4 без остатка, в другую — те числа, при делении которых на 4 получается остаток 1, и т. д.

$P$

55	144	9	3	71	35
46	93	85	10	80	24

158

Даны правила игры *Минусы*.

### Правила игры *Минусы*

Начальная позиция. В строке написано несколько минусов. Сколько именно, определяется дополнительными правилами.

Возможные ходы. На каждом ходу игрок переправляет один минус на плюс или два соседних минуса на два плюса.

Как определить победителя. Игра заканчивается, если очередной ход сделать невозможно — минусы закончились. Выигрывает тот, кто сделал последний ход.

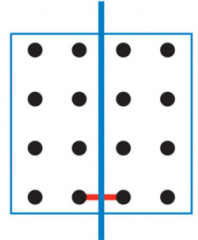
Найди выигрышную стратегию в игре *Минусы* с начальной позицией 18 минусов и в той же игре с начальной позицией 21 минус. Определи, кто обладает выигрышной стратегией, и опиши эту стратегию для каждой из данных начальных позиций. Теперь попробуй обобщить свои выводы — опиши выигрышную стратегию для любой игры *Минусы*:

- а) если начальная позиция — нечётное число минусов;
- б) если начальная позиция — чётное число минусов.



159

В игре *Ползунок* на поле  $4 \times 4$  равновесную выигрышную стратегию для Первого, которая описана на с. 99, построить не удастся. Но в этой игре существует другая равновесная выигрышная стратегия для Первого. Она получается, если считать равновесными такие позиции, в которых при перегибании поля по синей линии его правая и левая части совпадают. Эта стратегия заключается в зеркальном повторении ходов Второго (представь себе, что зеркало стоит на синей прямой).



Построй последовательность позиций какой-либо партии, в которой Первый следует такой стратегии и первым ходом соединяет две средние точки нижнего ряда.

160

Построй два таких множества бусин А и В, для которых все следующие утверждения истинны:

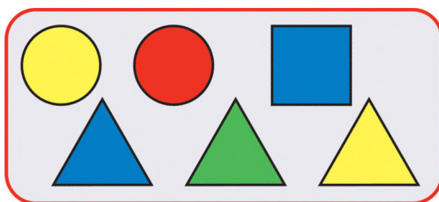
Множество П равно пересечению множеств А и В.

Множество О равно объединению множеств А и В.

В множестве А жёлтых бусин больше, чем квадратных.

В множестве В треугольных бусин больше, чем красных.

О



П



161

*Робот* находится на прямоугольном поле, внутри которого стен нет. Составь алгоритм, при выполнении которого *Робот* закрашивает все клетки поля, прилегающие к стенам.

162

Сколько разных чисел можно получить, переставляя цифры числа **5434**? Построй дерево перебора вариантов.

163

Исследуй игру *Оттесни шашку*. У кого из игроков есть равновесная выигрышная стратегия? Сформулируй эту стратегию.

### Правила игры *Оттесни шашку*

Начальная позиция. Полоска  $1 \times 20$  клеток. В крайних клетках полосы стоят белая и чёрная шашки.

Возможные ходы. Каждый игрок на своём ходу передвигает свою шашку на одну или две клетки по направлению к середине полосы, если это возможно. Перепрыгивать через шашку противника нельзя. Первый двигает белую шашку, Второй — чёрную.

Как определить победителя. Игра заканчивается, если очередной ход сделать невозможно. Выигрывает тот, кто сделал последний ход.

164

Найди выигрышную стратегию в игре *Камешки* (начальная позиция **308**, разрешается брать **1**, **2** или **3** камешка).



Для решения необязательно раскрашивать числовую линейку от 0 до 308 целиком, а можно:

- 1) раскрасить позиции от 0 до 16;
- 2) найти закономерность расположения проигрышных позиций на числовой прямой;
- 3) определить, какой будет начальная позиция, а значит, выяснить, кто из игроков обладает выигрышной стратегией;
- 4) сформулировать выигрышную стратегию, не перечисляя проигрышные позиции, а описывая их.

165

Построй последовательность однозначных чисел длины 5, для которой все следующие утверждения истинны:

В этой последовательности следующее число после каждого нечётного — чётное.

3

4

1

К

Первый член этой последовательности больше третьего на 2.

Каждый член этой последовательности есть в множестве К.



Даны правила игры *Две кучи камешков 2*.

### Правила игры *Две кучи камешков 2*

Начальная позиция. Две кучи камешков (сколько камешков в каждой куче, устанавливается дополнительными правилами).

Возможные ходы. На каждом ходу игрок может взять либо сколько угодно камешков из одной кучи, либо поровну камешков из обеих куч одновременно.

Как определить победителя. Игра заканчивается, если все камешки закончились. Выигрывает игрок, который забрал последний камешек.

Напиши последовательность позиций партии игры *Две кучи камешков 2*:

- а) с начальной позицией (9; 6), в которой выиграл Первый;
- б) с начальной позицией (7; 4), в которой выиграл Второй.

Найди выигрышную стратегию в игре *Две кучи камешков 2* с начальной позицией (6; 10) и в той же игре с начальной позицией (9; 8):

- 1) раскрась таблицу  $11 \times 11$ , начиная с заключительной позиции — клетки (0; 0);
- 2) определи, какой будет каждая из данных начальных позиций — выигрышной или проигрышной, а значит, у кого из игроков есть в этой позиции выигрышная стратегия;
- 3) сформулируй выигрышную стратегию для каждой из данных начальных позиций.

Теперь для каждой из данных начальных позиций запиши последовательность позиций какой-нибудь партии, в которой один из игроков использует выигрышную стратегию, а другой на каждом ходу берёт по одному камешку из каждой кучи.

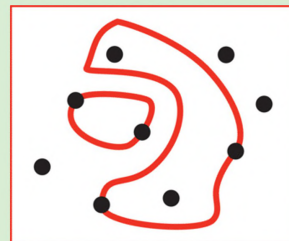
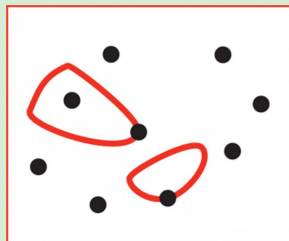
*Робот* находится в тупике (в закрытом конце) прямого коридора шириной в 1 клетку, идущего в неизвестном направлении. В конце коридора есть выход. Составь алгоритм, выводящий *Робота* из этого коридора.

Английский математик Джон Хортон Конвей (род. 1937) придумал много интересных математических игр. Вот правила одной из таких игр:

### Правила игры *Ободок*

Начальная позиция. Несколько точек на плоскости.

Возможные ходы. На каждом ходу игрок может провести одну замкнутую кривую (ободок) либо через одну из точек, либо через две. При этом два ободка пересекаться не должны. Вот примеры разрешённых ходов:



Как определить победителя. Игра заканчивается, если очередной ход сделать невозможно — все точки лежат на ободках. Выигрывает тот, кто сделал последний ход.

Известно, что в игре *Ободок* с любой начальной позицией у Первого есть равновесная выигрышная стратегия. Найди равновесную выигрышную стратегию для этой игры с начальной позицией 7 точек и для той же игры с начальной позицией 8 точек. Как в каждой из этих двух начальных позиций Первый должен провести ободок, чтобы сделать позицию равновесной? (Своим первым ходом Первый должен разделить все точки на две части так, чтобы каждый ход, сделанный в одной части, можно было повторить в другой.) Сформулируй выигрышную стратегию для каждой из данных начальных позиций. Попробуй обобщить своё решение — построй выигрышную стратегию для игры *Ободок*: а) если в начальной позиции нечётное число точек; б) если в начальной позиции чётное число точек.



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.

		A						B
						C		
D								
			E					F
						G		H

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

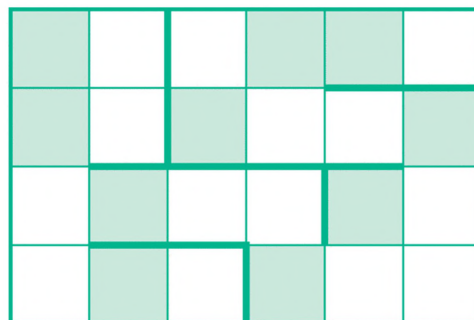
Для каждого составного условия с частицей «не» напиши простое условие с таким же значением истинности — построй и заполни таблицу в тетради.

№	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

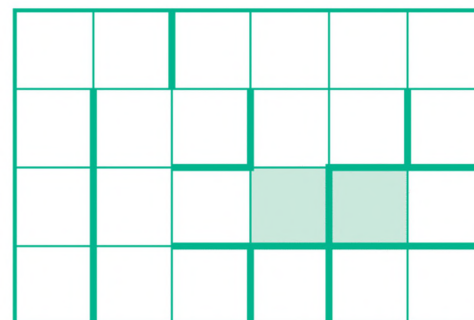
Нарисуй такое же поле в тетради, напиши в каждой клетке номер составного условия, которое истинно для этой клетки. При этом, конечно, в клетке может стоять несколько номеров.



№	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

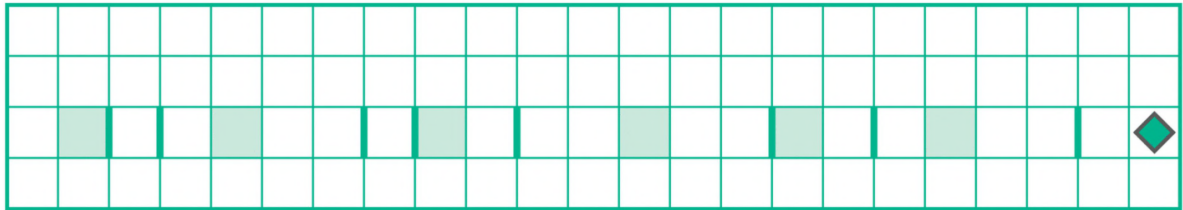
174

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							
	А	В	С	Д	Е	Ф	Г	Н
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
  
```

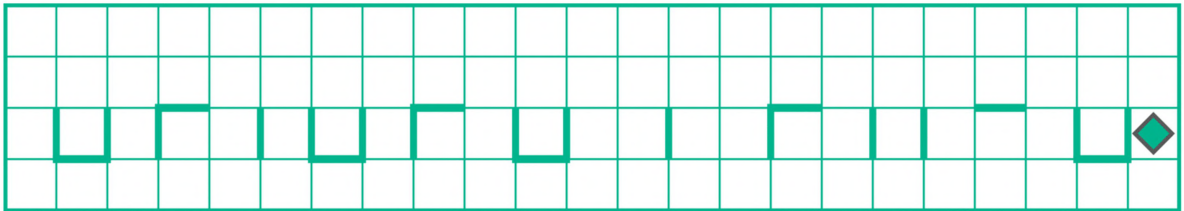
- 176** Найди все возможные числа, в которые может попасть *Кузнецик* с системой команд вперёд 3, назад 2, после выполнения им не больше 15 команд из начального положения «100».
- 177** Вот грузинские буквы. Найди три одинаковые буквы, напиши такую букву в тетради. Найди букву, которая встречается здесь 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**

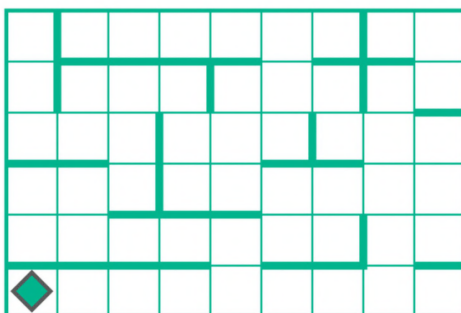
**кц**

**кон**

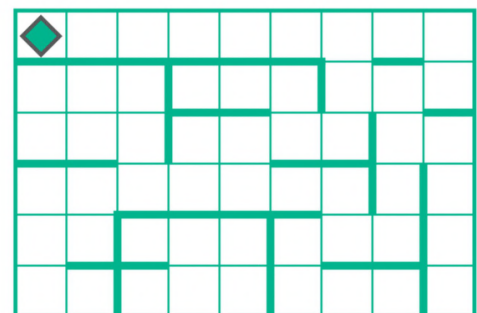
179

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 наклейте заготовку поля, раскрасьте клетки, нарисуйте 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.



# Биоинформатика.

## Белки и ДНК. Почему дети похожи на родителей?

С древних времён люди замечали, что у тигров рождаются тигрята, у птиц — птенцы, у людей — дети. Мало того, дети обычно внешне похожи на родителей — имеют тот же цвет глаз, цвет волос или форму носа. Новорождённый младенец часто не похож ни на мать, ни на отца, но со временем он приобретает черты внешнего сходства с матерью и отцом. Получается, что любой организм уже при рождении (а на самом деле ещё до рождения) «знает», какие у него во взрослом состоянии будут глаза, рост или голос. Значит, вся эта информация уже заложена с рождения, она где-то хранится, а по мере роста организм считывает эту информацию и приобретает те черты, которые ему предписаны наследственной программой.

Где же и как хранится наследственная информация? Как организм считывает её и понимает? Как он использует эту информацию по мере развития и роста? Этим вопросам много веков. Но ответы учёные стали находить только в последние 50 лет.

Всё в мире состоит из отдельных частей — и предметы, и живые организмы. В свою очередь, эти части сами состоят из частей и т. д. Некоторые более крупные части мы видим, а есть такие



Почему дети похожи на родителей?



маленькие, которые наш глаз не способен различить (они в миллионы раз тоньше человеческого волоса). Мы видим, например, что дом сложен из кирпичей, но мы не можем различить, из каких частичек состоит вода в стакане. И в воде, и в кирпичах, из которых построен дом, и в клетках тела человека можно выделить мельчайшие частицы, которые называются *молекулами*. Всё вокруг нас — и живое, и неживое — построено из молекул. В разных веществах молекулы различны. Они очень малы, их невозможно увидеть без специальных приборов. К счастью, такие приборы существуют — они помогают изучать молекулы.



Среди молекул любого живого организма на нашей планете основную часть составляют молекулы особых веществ — **белков**.

Различных белков, т. е. различных видов молекул белков, очень много — несколько миллионов. Например, только в организме человека встречается около 30 000 различных белков. Но при этом молекулы всех белков устроены похожим образом.



Молекула любого белка — это **цепочка (последовательность)**, состоящая из сотен, а иногда и тысяч звеньев. При этом во всех известных молекулах белков **встречается только 20 видов звеньев!** (Можно сказать, что книга жизни написана в 20-буквенном алфавите.)

Звенья белковых цепей называют *аминокислотными остатками*. Каждый аминокислотный остаток имеет своё название и обозначение (одной буквой латинского алфавита), поэтому белки часто описывают словами (последовательностями



Компьютерная модель молекулы белка. Радужная расцветка позволяет проследить ход звеньев цепочки. Молекула на рисунке увеличена примерно в миллиард раз



букв) в 20-буквенном алфавите (см. таблицу на форзаце в конце учебника).

Именно набором белков один организм отличается от другого. Наборы белков у двух людей (у двух кошек, у двух берёз) очень похожи, только мелкие различия определяют, например, разный цвет глаз у разных людей или разную расцветку шерсти двух кошек. Наборы белков у организмов разных видов разные, но чем более родственны эти виды, тем более похожи наборы белков. Например, белки человека и шимпанзе совпадают на 99 % (т. е. различается только одно звено из 100). А у человека и мыши степень сходства около 80 % (различаются 20 звеньев из 100).

Где же хранится наследственная информация?



За хранение и передачу наследственной информации в живых организмах отвечают специальные молекулы — **молекулы ДНК**.

ДНК — это сокращение, полное название — дезоксирибонуклеиновая кислота. Молекулы ДНК во всех клетках одного живого организма одинаковы. Но при этом молекулы ДНК разных организмов разные: у каждого человека свои молекулы ДНК, у каждой мышки свои.

Все молекулы ДНК, как и молекулы белков, — это цепочки, но звенья в молекулах ДНК отличаются от звеньев белков.



Звенья ДНК называются **нуклеотидами**. В молекулах ДНК встречается всего 4 вида нуклеотидов.

Молекулы ДНК в клетках живых организмов гораздо длиннее молекул белков. Даже самые короткие молекулы ДНК (ДНК вирусов) содержат сотни тысяч звеньев (нуклеотидов). А ДНК человека содержит около трёх миллиардов нуклеотидов. То есть молекула ДНК — это целая книга, написанная в 4-буквенном алфавите. Нуклеотиды обозначаются латинскими буквами А, С, G и Т.



Молекулы ДНК каждого живого организма **полностью определяют**, какие белки будут в этом организме.

Но как именно это происходит? Иными словами, как в ДНК закодированы (т. е. зашифрованы) белки? На этот вопрос мы ответим позже. А пока займёмся просто шифрованием.

# Шифрование

С древних времён люди использовали шифрование для секретной передачи и хранения информации. Шифрование выглядит как увлекательная игра, но преследует серьёзные цели. Шифры используются в военных целях, для передачи секретных сообщений, для хранения тайного знания и во многих других случаях.

Первые зашифрованные сообщения использовались ещё в Древнем Египте. Способ шифрования тогда был очень прост, сейчас он называется «шифрование простой подстановкой»: каждый иероглиф исходного сообщения заменялся в зашифрованном сообщении другим. При этом одинаковые иероглифы заменялись одинаковыми, а разные — разными.

Сегодня существует много способов шифрования и *шифров*. Мы будем пользоваться лишь одним способом, который построен по следующим правилам:

1. Каждая русская буква, а также пробел или знак препинания заменяется последовательностью латинских букв длины 3. Такая последовательность называется **кодом**. При этом используются только четыре латинские буквы — A, C, G, T.
2. Каждый код всегда заменяет (**кодирует**) одну и ту же букву или знак. При этом одна буква или знак необязательно всегда заменяется одним и тем же кодом.
3. При замене букв и знаков кодами порядок букв и знаков не меняется.



Замена каждой буквы её кодом называется **шифрованием**. Обратная замена каждого кода на соответствующую ему букву называется **расшифровкой**.

Для удобства шифрования и расшифровки строят *шифровальную таблицу*, в которой указан код (или все коды, если их несколько) для каждой буквы, для пробела и знаков препинания. Незаполненная часть такой таблицы помещена ниже.



Буква/знак	Код
А	
Б	
В	
Г	
Д	
Е	
Ё	



**Полный шифр** — это заполненная шифровальная таблица, указывающая соответствие каждой буквы или знака и каждого кода.

Вот примеры шифровок:

Код буквы Я — САТ.

Шифровка слова ТЫ — ССССГГ.

Зашифруем слово ОНИ — получим шифровку АСТАГСТАА.

Закодируем слово КОМПЬЮТЕР — получим шифровку AGGACTACGATCSTTCGACCCAGAAGT.

После раскодирования шифровки GAAAAAСТСТААССGAGT-АСТАAGAAAAGCAGCCGGACC получаем слово ЗАШИФРОВАННЫЙ.

Обратите внимание, что знак переноса нам помогает удобно расположить на странице слова и их шифровки. При этом знак переноса при шифровании и расшифровке не учитывается.

Полный шифр вы постройте, решая задачи.

**182**

Вырежи из вкладыша тетради проектов заготовку шифровальной таблицы и вложи в тетрадь (лучше прикрепить её скрепкой, чтобы потом не выпала). Используя коды из примеров на этой странице выше, заполни в своей таблице все строки, которые сможешь.



Проверь: в таблице должны быть коды для 19 букв.

183

Зашифруй те слова множества F, для шифрования которых в твоей таблице имеются все необходимые коды. Запиши шифровки в тетрадь.

ШАР	БАНАН	ШАРФ	КАРТОШКА
ОНА	ОН	ВЬЮНОК	ДЕДУШКА

F

184



Сколько различных последовательностей длины 3 можно составить из букв множества M (конечно, буквы могут повторяться)?

Построй дерево перебора вариантов. Можно ли было использовать для шифрования букв русского алфавита не тройки, а пары, составленные из букв множества M? Поясни свой ответ.

A	T
C	G

M



При расшифровке удобно пользоваться обратной шифровальной таблицей — *таблицей расшифровки*. В такой таблице в левой колонке выписаны все возможные коды (в словарном порядке), а в правой — буквы или знаки, которые этим кодам соответствуют. В заготовке таблицы расшифровки заполни все строки, которые сможешь.

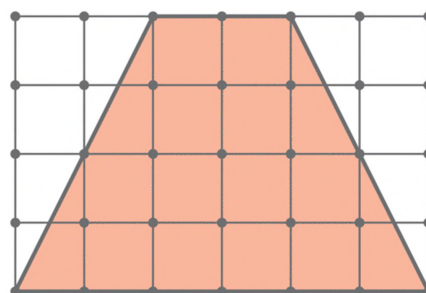
185

Раскодируй шифровки: перепиши их в тетрадь и напиши рядом с каждой шифровкой зашифрованное в ней слово.

A	G	G	A	C	T	G	A	A	A	A										
A	G	G	A	C	T	G	A	A	C	G	G									
A	T	C	A	G	A	A	G	T	A	A	G	A	A	C	A	T				
A	A	A	A	G	C	C	C	C	A	G	T	A	A	A	A	G	G	C	C	C

186

Нарисуй, как разрезать четырёхугольник на части, чтобы из этих частей можно было собрать прямоугольник на сетке. Нарисуй этот прямоугольник и покажи штриховыми линиями, из каких частей он составлен.





187

Множество В — множество шифровок всех слов из множества А. Запиши для каждого слова его шифровку и заполни пустые клетки шифровальной таблицы и таблицы расшифровки.



Проверь себя — в каждой таблице теперь должны быть коды для всех русских букв.

А

УХ	ЭХ
ЧАС	УЖ
	БАЦ
БАС	
	ЛУГА
ДУГА	
	ЛУЖА
ОБЩИХ	
ОБЪЁМ	

А

В

CAGCCT	AACAAACAC
CCACCT	AACAAAATG
CCACAA	CGCAAAATG

В

ACACCAAATAAA  
ATTCCACAAAAA  
ATTCCAAATAAA

АСТААСГССТААССТ  
АСТААСТССАТААСГ

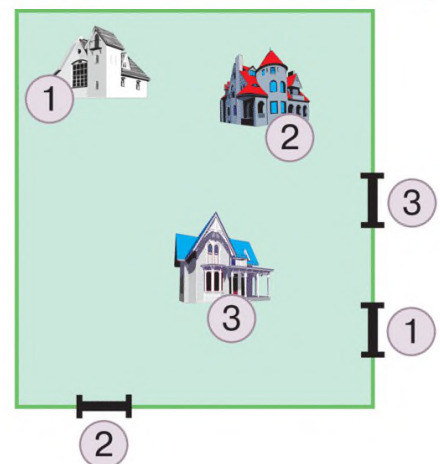
188

Дано зашифрованное предложение. Слова в этой шифровке разделены тройкой символов, кодирующей пробел. Расшифруй и запиши в тетрадь предложение. Дополни таблицы шифровки и расшифровки кодом пробела.

А С G А А А С G А А А С Т А А С G С G G  
А Т Т А А А С Т А А G Т А А А А С G С С А

189

На квадратном участке расположены три дома, а в ограде сделаны три калитки. Проложи дорожку от каждого дома к калитке с тем же номером так, чтобы дорожки не пересекались. Нарисуй схему участка и дорожек в тетради. Дома перерисовывать не надо — достаточно поставить номера.



190

При помощи таблицы расшифровки раскодируй следующие шифровки, запиши получившиеся слова.

АТСААААГТААААСААСТАГГАТГ  
 ССГАГАААГАГТААААТТСТТ  
 АСГССАГААСГГАГГААА  
 АСГАГААСТСССТССАГААСГ-  
 АТТАГААСГСГГАСС

191

При помощи шифровальной таблицы зашифруй слова: ПАРОМ, ВОЗДУХ. Теперь, не обращаясь к шифровальной таблице, зашифруй слова: ПАРОВОЗ, ДУХОМ.

192

Пользуясь шифровальной таблицей, зашифруй предложение: ЛЮБЛЮ ГРОЗУ В НАЧАЛЕ МАЯ.

193

Раскодируй зашифрованное предложение.

АТСАГТТААССТАСТАСАТ-  
 ААСТААТГАГАААТАСТАС-  
 ААГССАТСТААГСАААСТА-  
 АГГАААСССАСТАГГСТАА-  
 АГСТААТГАГААСГСТТ

194

Найди выигрышную стратегию для игры *Двадцать пять*.

### **Правила игры *Двадцать пять***

Начальная позиция. Число 0.

Возможные ходы. На каждом ходу игрок прибавляет к имеющемуся числу 1, 2, 3 или 4.

Как определить победителя. Игра заканчивается, если позиция оказывается равной 25. Выигрывает тот, кто добавил последнее число.



# Биоинформатика.

## Как кодируются белки



Раскрашенная модель молекулы ДНК. Молекула ДНК очень длинная — здесь представлена только небольшая её часть. На модели видно, что молекула ДНК состоит из двух цепочек (одна раскрашена жёлтым, а другая — красным). В этих цепочках нуклеотиды расположены друг против друга и связаны особыми химическими связями (они показаны голубым)

Ты уже знаешь, что в состав любого живого организма входят молекулы белков. Именно набор белков определяет, например, почему у одного человека глаза карие, а у другого — голубые. Каждая молекула белка — это цепочка (последовательность). Звенья этой цепочки называются *аминокислотными остатками* или просто *остатками*. Все белки каждого живого существа закодированы в особой молекуле — молекуле ДНК. Молекула ДНК — это тоже цепочка, состоящая из звеньев другого вида — *нуклеотидов*. Таким образом, в одной последовательности (молекуле ДНК) закодированы другие последовательности (молекулы белков). А как именно, каким шифром в ДНК закодированы белки?

Оказывается, это происходит примерно так же, как в задачах на шифрование, которые ты решал.

Аминокислотные остатки (звенья молекул белка) могут быть только двадцати видов. Это значит, что молекула белка похожа на длинное слово, написанное в 20-буквенном алфавите. Каждый из двадцати возможных остатков имеет своё название и обозначается одной латинской буквой (см. таблицу на форзаце в конце учебника).



Все молекулы ДНК построены только из четырёх видов нуклеотидов. Вот их русские и английские названия и буквы, которыми они обозначаются:

аденин (*Adenine*, A),  
цитозин (*Cytosine*, C),

гуанин (*Guanine*, G),  
тимин (*Thymine*, T).

Молекулу ДНК можно сравнить с очень длинным словом, написанным в 4-буквенном алфавите.

Не вся молекула ДНК кодирует белки, а только некоторые её участки, которые называются **генами**. Молекула ДНК простейших организмов (вирусов, бактерий) почти вся состоит из генов, а в молекуле ДНК человека гены составляют только около 3 % всей длины. Зачем нужны остальные 97 % ДНК человека, науке пока известно не очень хорошо.

В гене (как и в наших шифровках) каждый остаток белка кодируется тройкой нуклеотидов. Такие тройки биологи называют *кодонами*. Например, тройка ААГ (аденин — аденин — гуанин) кодирует остаток лизин. При этом один и тот же остаток может кодироваться разными тройками нуклеотидов. Специальные тройки кодируют начало и конец гена. В таблице на форзаце в конце учебника вы найдёте полный список остатков и всех шифрующих их кодонов. Эта таблица называется *таблицей генетического кода*.

О том, что белки могут кодироваться тройками нуклеотидов, догадался в середине XX века выдающийся физик и биолог Георгий Гамов. Позже с помощью экспериментов учёные подтвердили эту догадку и определили, какой остаток кодируется каждым кодоном. Замечательно, что *генетический код един для всех известных живых организмов!*

195

В этой шифровке закодировано слово, но по ошибке то ли одну букву пропустили, то ли вставили лишнюю. Не расшифровывая слово, определи, какую именно ошибку допустили при шифровании: пропустили букву или вставили лишнюю. Объясни свой ответ.

А А А А А С А С А Г Т А Г А А А Г Т-  
А А А А А С С С С С А А Г Т А А А



196

Одно из слов множества Д зашифровали, но при этом допустили ошибки — одну букву в шифровке пропустили и вставили одну лишнюю (необязательно в то же место). Определи, какое слово пытались зашифровать, запиши в тетрадь это слово и его правильную шифровку.

С С А А Т-  
А А G G

ТОК ТАК ШОК  
ШИК ТИК

Д

197

Разведчик зашифровал два слова — важный пароль и ответ на него. В начале и в конце, а также между словами разведчик вставил лишние тройки латинских букв, которые не входят в послание, — они служат для того, чтобы замаскировать эти важные слова в тексте (на случай перехвата послания противником). При этом начало и конец каждой части послания разведчик закодировал специальными тройками латинских букв, чтобы в штабе послание всё же смогли расшифровать. Этими тройками разведчик всегда кодирует начало и конец предложения. Расшифруй и запиши в тетрадь пароль и ответ на него. Дополни таблицы шифровки и расшифровки тройками, кодирующими начало и конец предложения.

T T G G C A C G T A T C A G A C C T-  
A C T C C C A A A C T G T C T G A T-  
C G T A A A A A G T A A A A A C A C-  
T A A C A T C T G G T T G G A

198

Реши задачу.

Известно, что К. М. Петров, В. Д. Петров, П. Б. Петров, Н. В. Петров, М. С. Петров, И. В. Петров, А. К. Петров, Д. М. Петров, Р. Б. Петров, Г. Д. Петров, Б. К. Петров — представители одного рода, причём один из них — основатель рода, остальные — его сыновья, внуки и правнуки. Других сыновей, внуков и правнуков у основателя рода не было. Построй дерево родства Петровых, если известно, что у каждого отца было по два сына, внуков у основателя рода — четыре, а у его сыновей — по два.

199

Дано три шифровки одного слова с ошибками. В одной пропустили одну букву, в другой вставили лишнюю, в третьей произошло и то и другое. Запиши слово и его правильную шифровку.

```

A C A G T C C C A A C A
A T C A G G T C C A A C A
A T C A G T C A A C A

```

200

Наиболее важные послания разведчик шифрует более сложным шифром, чем обычно. Он использует те же коды для тех же букв, что и раньше, но использует также и новые, дополнительные коды (тоже тройки букв из того же 4-буквенного алфавита). Таким образом, одну и ту же русскую букву он может шифровать разными кодами (при этом один и тот же код, как обычно, всегда кодирует одну и ту же букву). Множество W состоит из разных шифровок двух слов: СМЕШНО и ГЛАДИТ. Выясни, какая шифровка к какому слову относится, дополни новыми кодами букв свою шифровальную таблицу (в ней станет 52 кода) и таблицу расшифровки.

```

GGAGTTGAGGGCGTGTGT
TCTGCAGGTTTCAGTATTA
AATATTGACGGCGTGCCC
TGCTTTGCTTCAAGCTTC
GGAGTTGATGGCTACTGT

```

W

201

Молекула ДНК человека состоит из трёх миллиардов нуклеотидов. При этом гены составляют только 3 % всей длины последовательности. Сколько нуклеотидов содержится во всех генах человека?

202

На поле *Робота* стен нет. *Робот* находится в левом верхнем углу прямоугольника из закрашенных клеток. Составь алгоритм, переводящий *Робота* в правый нижний угол прямоугольника.



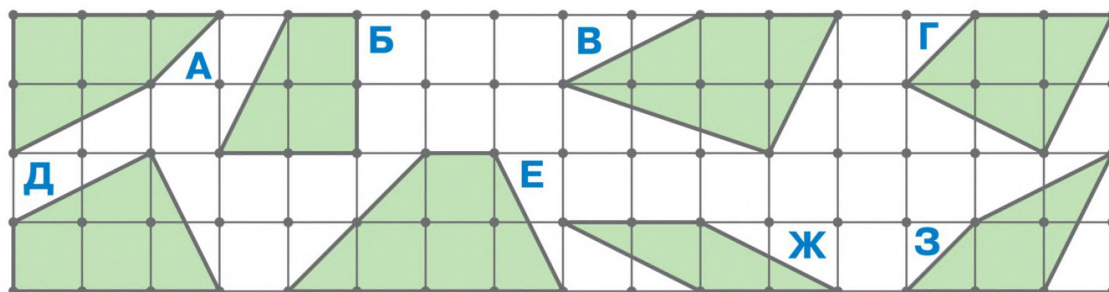
203

Разведчик зашифровал два предложения и вставил лишние тройки латинских букв, которые не входят в послание, в начале, в конце и между предложениями. Расшифруй послание.

G C G T C T C G T A T G C C C C A -  
 T A A T T A A A A G A A A A G C T A -  
 A A G A C T A A A G A C T A C C A T -  
 G A G G C T A A G C A A A C T A G A -  
 A A A A A T C A A A A C A C T A A C T -  
 A A C A C G A A A A G C C T G T A T -  
 A G T G G G C G T A A C A G A A G T -  
 A G A A A T T A A C C C A G A C T A -  
 C G A C A A A G C C G G A G A C T A -  
 A A T A G T A A A A G C T A A C A C -  
 C G G C T G T T T G T A T T C G G T

204

Найди два четырёхугольника одинаковой площади.



205

Участок ДНК

ATG CCA GCC ACA GAC ACA  
 AAC AGC ACC CAC ACC ACG CCG  
 ATG CAC CCA GAC GCC CAA CAC

кодирует такую последовательность остатков:

M P A T D T N S T H T T P M H P D A Q H

Пользуясь таблицами на форзаце в конце учебника, найди такой остаток, который в этом примере кодируется тремя разными кодонами. Выпиши его название и его кодоны.

206

Дано множество R слов и множество Q шифровок этих слов. Напиши рядом с каждым словом его шифровку. Дополни новыми кодами букв шифровальную таблицу и таблицу расшифровки (должен быть 61 код).

**R**

ПАР	
	ВОР
ЗРЯ	
	УХО
РОК	
	ПОЛ
КОК	

**Q**

GGGTTATAT
TATACTTGG
GCGTATTTCG
TGATTTCGTT
TTGGAGTAT
GTCTTATGG
TAGCCTTTC

207

Реши задачу.

Каждый из четырёх гномов — Ваня, Даня, Женя и Саня — либо всегда говорит правду, либо всегда врёт. Мы услышали такие отрывки разговора: Ваня — Дане: «Ты врун». Даня — Жене: «Ты врун!» Женя — Сани: «Оба они вруны». Потом Женя подумал и добавил: «Да и ты тоже врун». Кто из этих четырёх гномов всегда говорит правду?



208

Реши задачу.

При приготовлении пиццы выпекается большой хлебный корж и посыпается тёртым сыром. К сыру добавляются разные продукты, обеспечивающие тот или иной вкус. В распоряжении Марчелло имеются сладкий перец, репчатый лук, маринованные грибы, свежие помидоры, маринованная морковь и анчоусы (мелкая рыбка — хамса специального посола). По мнению Марчелло, имея один или несколько из этих продуктов (в любых сочетаниях), а также корж и сыр, можно приготовить пиццу. Сколько типов пиццы можно приготовить?



# Автомат-сортировщик

Теперь мы познакомимся с методом половинного деления, анализируя игру *Угадай число*.

Но сначала поговорим о сортировках и роботах-сортировщиках.

Представим себе Автомат, который умеет сортировать элементы множеств. Будем называть его Автомат-сортировщик. Наш Автомат-сортировщик на каждом этапе сортировки разделяет каждое множество на две непустые части. На первом этапе Автомат разделяет исходное множество на две части, на втором разделяет каждую из двух полученных частей ещё на две и так далее, пока в каждой части не останется только один элемент. Если Автомату попадается множество, состоящее только из одного элемента, то он с таким множеством ничего не делает — оставляет его как есть. Для удобства назовём такой Автомат-сортировщик *Простым автоматом*.

Пусть дано множество  $A$  чисел от 11 до 20.

**A**

11	13	15	17	19
12	14	16	18	20

Дадим *Простому автомату* задание рассортировать это множество. На следующей странице приведены два примера деревьев такой сортировки.

В сортировке с деревом  $K$  было 5 этапов, в сортировке с деревом  $L$  — 8 этапов. Как видите, сортировка *Простым автоматом* одного и того же множества может быть разной — множество  $A$  можно рассортировать и в 5 этапов, и в 8 этапов (можно даже в 9 этапов!).

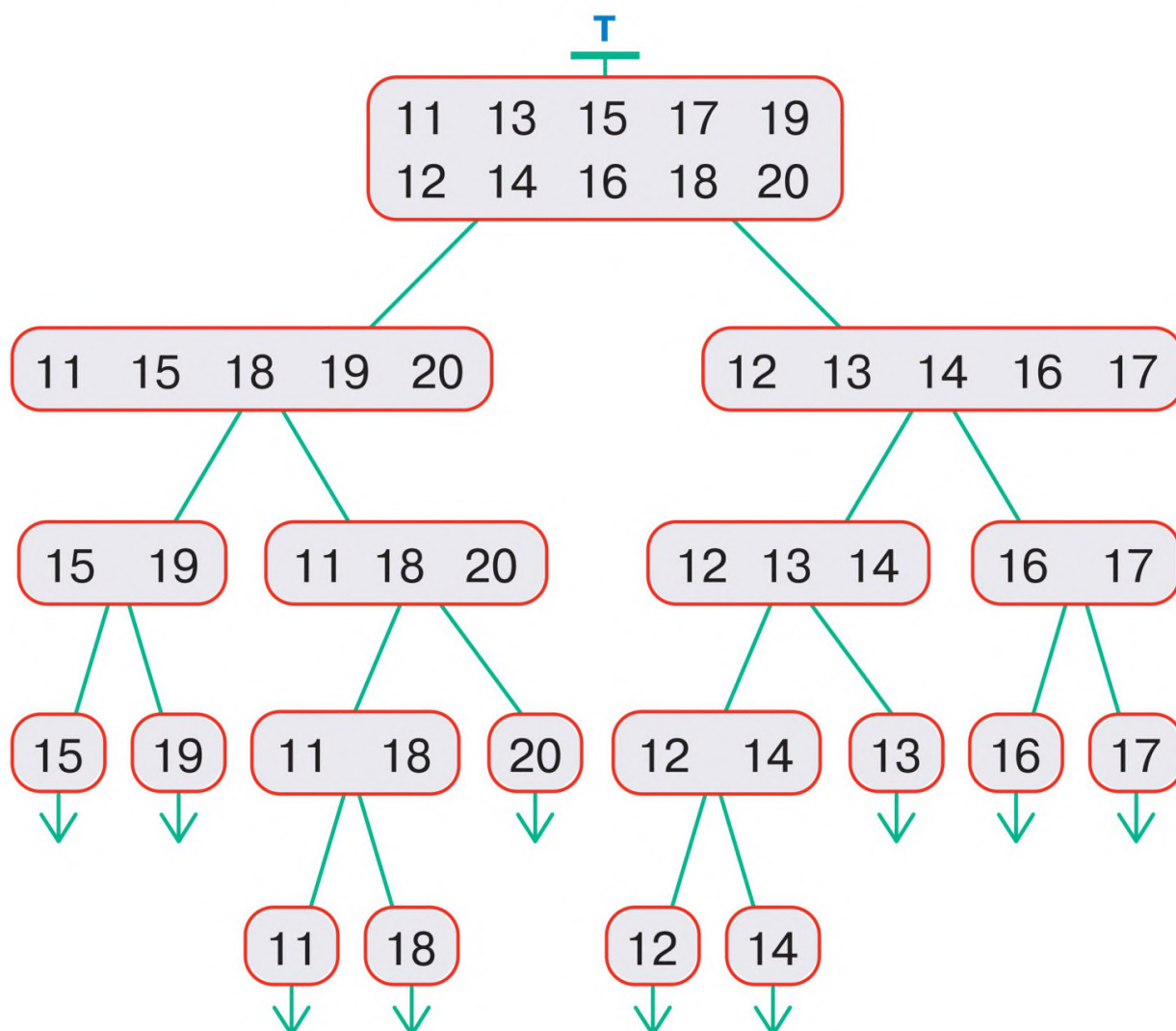
Давайте теперь усовершенствуем *Простой автомат*. Пусть теперь Автомат-сортировщик не просто разделяет множество на две части, а всегда разделяет его *примерно пополам*. Если в исходном множестве было чётное число элементов, то в множествах, полученных после деления, элементов будет поровну, а если в исходном множестве было нечётное число элементов, то в полученных множествах число элементов будет





различаться на 1. Такой усовершенствованный Автомат-сортировщик мы назовём *Половинным разделителем*.

Вот пример дерева сортировки *Половинным разделителем*:



При всех способах сортировки одного и того же множества *Половинным разделителем* число элементов дерева сортировки будет одним и тем же. Никакой Автомат-сортировщик не сможет рассортировать элементы множества за меньшее число этапов, чем *Половинный разделитель*.

**209**

Нарисуй какое-нибудь дерево сортировки *Половинным разделителем* множества всех гласных букв русского алфавита. Сколько уровней в твоём дереве? Сколько этапов в сортировке по этому дереву?

210

Вот описание игры *Угадай число*.

В игре *Угадай число* участвуют двое игроков: Водящий и Угадывающий. Водящий загадывает любое число, обычно от 1 и до заранее определённого числа. Угадывающий должен отгадать это число, задавая Водящему вопросы, на которые можно ответить только «Да» или «Нет».

Вопросы можно задавать самые разные, например, не хочется ли Водящему съесть яблоко. Можно просто перебирать числа: «Это число 1? Это число 2? и т. д.» Но так игра затянется надолго. Поэтому нужно установить, за какое число вопросов отгадывается число.

Сыграйте с соседом по парте четыре партии в игру *Угадай число* с такими правилами: разрешается загадывать число от 1 до 16, разрешённое количество вопросов — четыре. Если за четыре или меньше вопросов Угадывающий называет число, игра заканчивается и выиграл Угадывающий. Если после четвёртого вопроса число не отгадано, то партия закончилась выигрышем Водящего. Заполни таблицу соревнования.

Игроки Партии	Твоё имя	Имя соседа
1-я партия		
2-я партия		
3-я партия		
4-я партия		

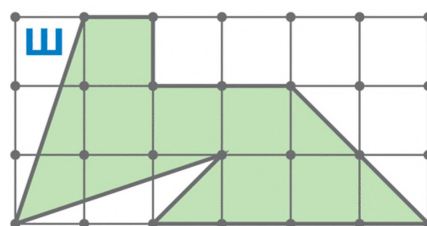
211

Реши задачу.

В магазин привезли 223 л молока в бидонах по 10 л и по 17 л. Сколько было бидонов?

212

Найди площадь данного многоугольника Ш.

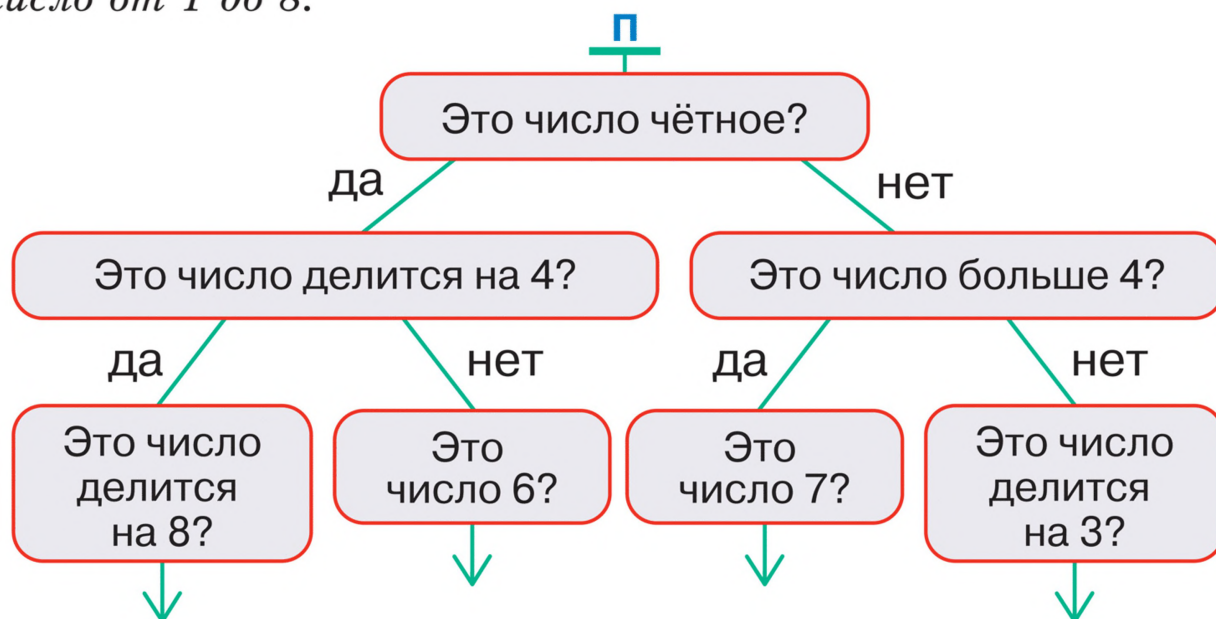




# Метод половинного деления

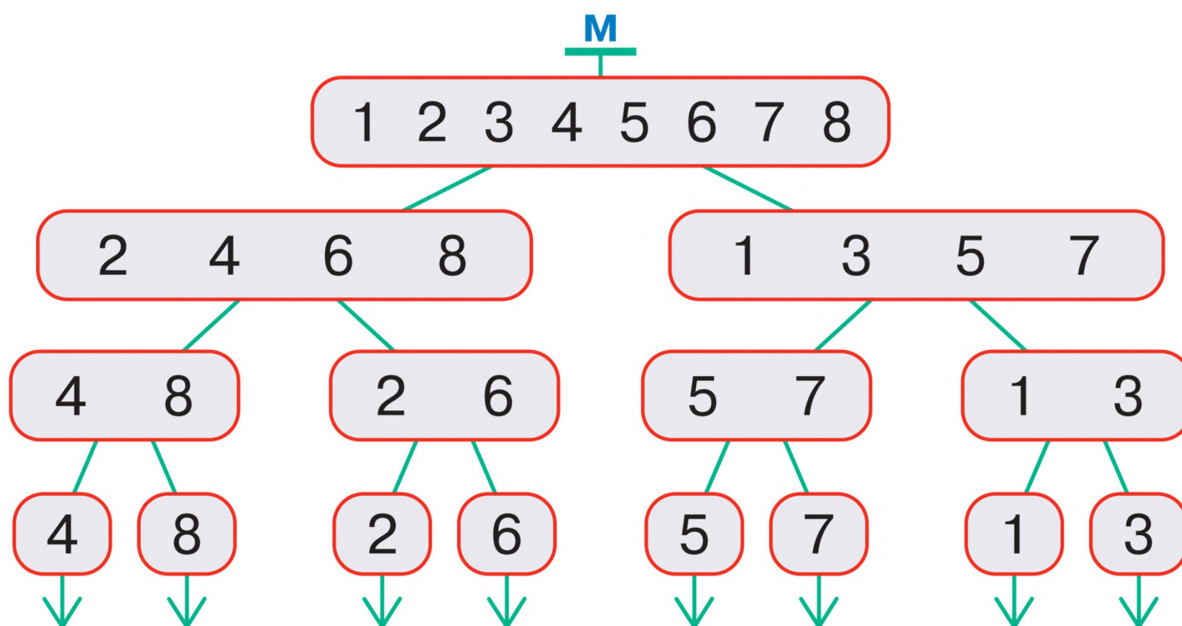
Подумаем, как играть в игру *Угадай число* (правила игры см. в задаче 210), чтобы наверняка угадать число за заданное число вопросов. Так как мы хотим угадать число наверняка, то нам придётся спланировать игру для любого задуманного числа, т. е. придумать все возможные вопросы, которые мы зададим Водящему в зависимости от его ответов. Сложность в том, что заранее составить список вопросов здесь невозможно — ведь каждый следующий вопрос зависит от того, как ответил Водящий на предыдущий вопрос — «Да» или «Нет». Поэтому придётся строить дерево вопросов. После каждого вопроса должны следовать два: один вопрос задаётся в случае ответа «Да», другой — в случае ответа «Нет».

Рассмотри пример такого дерева вопросов для игры *Угадай число от 1 до 8*.



Обрати внимание, что ответ на любой правильно поставленный вопрос разделяет исходное множество чисел на два множества: одно, для чисел которого на данный вопрос правильно ответить «Да», другое, для чисел которого на данный вопрос правильно ответить «Нет». Значит, по каждому дереву вопросов можно построить дерево сортировки исходного множества чисел *Простым автоматом*.

Вот дерево сортировки *Простым автоматом* множества чисел от 1 до 8, построенное по дереву вопросов П:



Итак, построив дерево вопросов и соответствующее ему дерево сортировки *Простым автоматом*, можно наверняка отгадать число из заданного множества за заданное число вопросов. При этом число вопросов будет наименьшим, если дерево вопросов (и дерево сортировки) будет наименьшей высоты, а наименьшая высота получается при сортировке *Половинным делителем*. Значит, вопросы Угадывающего должны быть такими, чтобы ответы на них разделяли каждое множество *примерно пополам* — так, как разделяет *Половинный делитель*. Используя такие вопросы, Угадывающий всегда отгадает загаданное число, задав наименьшее количество вопросов.

Такой метод поиска элемента в заданном множестве называется **методом половинного деления**.

**213**



Используя метод половинного деления, построй дерево Ш вопросов в игре *Угадай число от 1 до 16*. Построй дерево Б сортировки *Половинным делителем* множества чисел от 1 до 16, соответствующее дереву Ш.



Сколько этапов в твоей сортировке? За сколько вопросов можно наверняка отгадать число от 1 до 16?

Пользуясь построенным деревом, сыграй с соседом две партии в игру *Угадай число от 1 до 16*.

214



Определи, за сколько вопросов можно наверняка отгадать число в игре *Угадай число от 1 до 25*. Чтобы ответить на вопрос, построй дерево сортировки *Половинным разделителем* множества чисел от 1 до 25.

Пользуясь построенным деревом, сыграй с соседом две партии в игру *Угадай число от 1 до 25* и проверь свои ответы.

215



Правила игры *Угадай букву* очень похожи на правила игры *Угадай число*. В игре *Угадай букву* Водящий загадывает букву русского алфавита.

Определи, за сколько вопросов можно наверняка отгадать букву в игре *Угадай букву*. Чтобы ответить на вопрос, построй дерево сортировки *Половинным разделителем*.

Пользуясь построенным деревом, сыграй с соседом две партии в игру *Угадай букву* и проверь свои ответы.

216

Сколько существует трёхзначных чисел, в запись которых входит ровно одна цифра 5?

217

Реши логическую задачу.

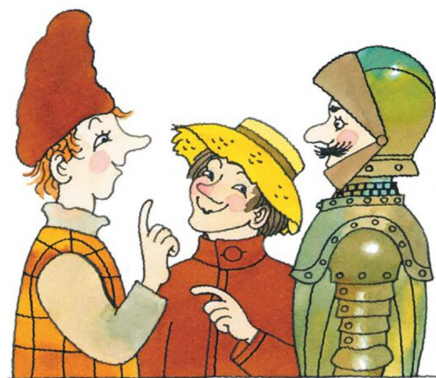
Жители двух соседних городов — города Рыцарей и города Лжецов — раз в год приезжают на ярмарку в город Хитрецов. Известно, что рыцари всегда говорят правду, лжецы всегда лгут, а хитрецы говорят правду через раз, т. е. из двух сказанных ими подряд предложений одно — правда, а другое — ложь. На ярмарке встретились трое из разных городов и затеяли спор:

Первый. *Один из нас рыцарь.*

Второй. *Да уж ты-то точно лжец.*

Третий. *Оба вы лжецы. Хотя я тоже не рыцарь.*

Определи, кто такие Первый, Второй и Третий, если точно известно, что все они живут в разных городах.



# Биоинформатика.

## Как изучают белки



На этой модели видно, что аденин (А) всегда связан с тиминном (Т), а цитозин (С) — с гуанином (G). Две цепочки молекулы ДНК закручены в спираль — наподобие винтовой лестницы. Эту модель (она называется «двойная спираль») придумали в 50-х гг. XX в. Дж. Уотсон и Ф. Крик

Ты уже знаешь, генетический код в основном устроен так же, как и код в наших задачах: есть текст в 4-буквенном алфавите (А, С, G, Т); тройки букв (тройки нуклеотидов молекулы ДНК) кодируют остатки — звенья молекул белков.

Участки ДНК, которые кодируют белки (они называются *генами*), ограничены специальными тройками. Первая тройка в любом гене — АТG (она называется *старт-кодоном*). Кодон АТG — это кодон остатка метионина (такой остаток обозначается буквой М), поэтому этот кодон встречается и внутри генов. Последняя тройка в каждом гене — это ТАА, TAG или TGA. Эти кодоны называются *стоп-кодонами*, они играют такую же роль, как точки в предложении. Стоп-кодоны не кодируют никакого остатка, внутри генов они не встречаются.

Ты уже знаком с *таблицей генетического кода*. В ней для каждого остатка приведены его кодоны, при этом один остаток, как правило, может кодироваться несколькими разными кодонами. На

форзаце в конце учебника приведена и обратная таблица: для каждого кодона указан соответствующий остаток.



Мы описали основной способ кодирования белков. Бывают и более сложные случаи: например, гены могут перекрываться, т. е. получается как бы одна шифровка, наложенная на другую. Такое явление встречается в ДНК вирусов. Похожие ситуации вам встретятся в задачах на шифрование.

Современные технологии позволяют определять последовательность нуклеотидов в молекулах ДНК. Сегодня уже полностью расшифрованы ДНК человека, шимпанзе, мыши, сотен других животных и растений и десятков тысяч бактерий. Теперь учёным важно понять, для чего служат различные фрагменты ДНК, и прежде всего выделить участки, кодирующие белки.

Иногда учёным приходится иметь дело с данными, в которых есть ошибки, — в ходе эксперимента некоторые буквы могут потеряться, а могут и ошибочно появиться новые буквы. Подобные задачи на шифрование тебе уже встречались. При их решении ты использовал то, что не все тройки допустимы. В реальном же кодировании белков это не так — каждая тройка имеет значение. Поэтому, чтобы выделять кодирующие участки, учёным-биологам приходится использовать более сложные методы.

218

Иногда в одной шифровке разведчику удаётся передать два сообщения, наложив «одно поверх другого». Для этого он так составляет шифровку, чтобы сначала читалось первое сообщение, а с некоторого места (например, со второй буквы шифровки) читалось второе сообщение. Вот пример такой шифровки, в которой содержатся с перекрытием 2 слова из 4 букв. Расшифруй и запиши в тетрадь эти слова, рядом с каждым словом запиши его шифровку.

С А А С G А А G Т Т А А А С А Т

219

Пользуясь шифровальной таблицей, напиши 6 разных шифровок слова МОЩЬ. Сколько всего существует таких шифровок? Объясни свой ответ.

220

Пользуясь своей шифровальной таблицей, определи, сколькими способами можно зашифровать слово ЖАДНЫЙ. Запиши все различные шифровки этого слова.

221

На форзаце в конце учебника приведена обратная таблица генетического кода — для каждого кодона указан соответствующий остаток. Обрати внимание, что главную роль в определении остатка, соответствующего кодону, играют первые 2 буквы кодона. В одних случаях третья буква может быть любой (так, например, для *пролина* годится любой кодон, первые 2 буквы которого — СС). В других — третьи буквы А или Г соответствуют одному остатку, а С или Т — другому. Например, ААА и ААГ кодируют *лизин*, а ААС и ААТ — *аспарагин*. Из этих правил есть только два исключения. Найди их в таблице генетического кода.

222

В одной из шифровок слова РЫСЬ вычеркнули 3 буквы (не обязательно идущие подряд), и получилась одна из шифровок слова ПЕЛ. Найди и запиши такую шифровку слова РЫСЬ, подчеркни в ней те буквы, которые вычеркнули, чтобы получить шифровку слова ПЕЛ.

223

Пользуясь своей шифровальной таблицей, придумай и запиши слово, которое можно зашифровать более чем 10 способами. Запиши любые 5 из возможных шифровок этого слова.

224

Запиши предложение, содержащееся в следующей шифровке (слова, как обычно, разделены тройкой латинских букв, кодирующей пробел):

```

T A G T A T A C T G T C C T A T T T -
C C A G C G C G G A G G T A C C T A -
T G A G G T A G T G C T G T A A C T T -
G C T A A C C C T C T T C G C T A A -
G C G A T C T A G A A A A A G G G C -
C C T A T G A C

```

225

Сколько разных чисел можно получить, переставляя цифры числа **9854**?



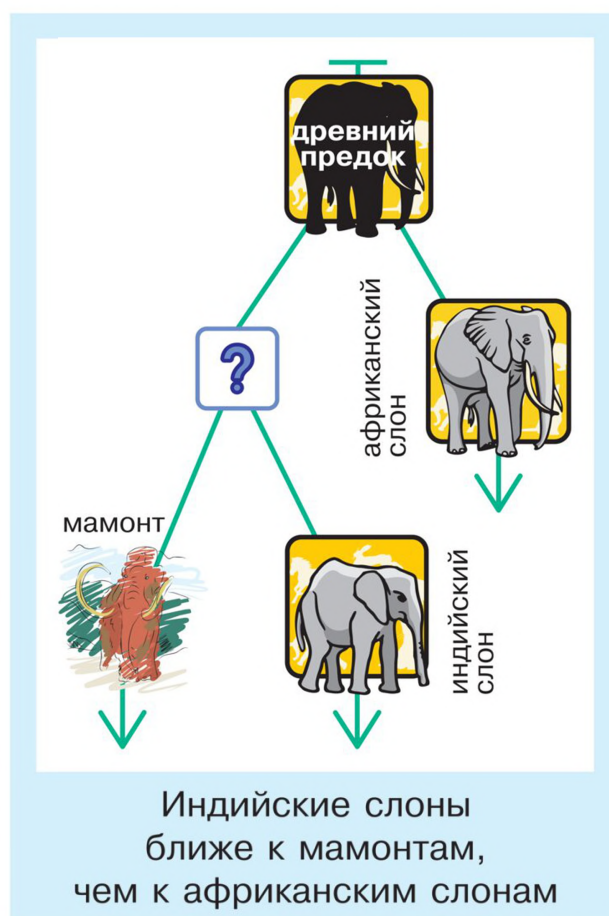
# Биоинформатика. Сравнение белков

У живых организмов различаются наборы белков. Наборы белков у двух людей (у двух слонов, двух бактерий) очень похожи, только мелкие различия определяют, например, различную форму уха у разных людей или разный цвет хвоста у двух кроликов.

Наборы белков у организмов разных видов разные, но чем более родственны эти виды, тем более похожи наборы белков. Например, сходство между белками человека и шимпанзе достигает 99 %.

Верно и обратное: если два вида организмов имеют похожие наборы белков, то эти виды являются родственными, т. е. происходят от какого-то одного вида древних животных. Пользуясь методами сравнения белков, учёные восстанавливают родственные связи и происхождение различных видов животных и растений. Например, недавно стало известно, что, судя по некоторым белкам, индийские слоны более близки к вымершим мамонтам, чем к современным африканским слонам.

Как учёные проводят сравнение белков? Как вы теперь знаете, белки — это последовательности аминокислотных остатков, их можно представить в виде длинных слов — последовательностей букв в 20-буквенном алфавите. Поэтому задачу можно сформулировать так: какие два слова считаются близкими, похожими, а какие — далёкими?



# Преобразование слов

Вы наверняка играли в игру, в которой одно слово надо превратить в другое, заменяя на каждом шагу только одну букву. Примерно такой игрой мы сейчас займёмся.

В отличие от тех игр, которыми мы занимались раньше, в эту игру можно играть одному. Цель игры — за наименьшее число шагов превратить одно слово в другое, соблюдая такое правило:

## Правило превращения слов

На каждом ходу можно выполнить одно из трёх действий:

- заменить одну букву на другую;
- вставить одну букву;
- удалить одну букву.

Конечно, по этим правилам одно слово можно превратить в другое разными способами. Например, из слова СЛОН можно сделать слово СОН за один ход, а можно и за 7 ходов. Последовательности таких превращений помещены снизу. Но цель нашей игры — найти самое короткое превращение, поэтому мы выбираем превращение R.

Давайте усложним игру — присвоим каждому ходу цену.

Ход	Цена
Замена гласной на гласную	1
Замена согласной на согласную	2
Замена гласной на согласную или наоборот	3
Замена Ъ или Ь на другую букву или наоборот	4
Вставка или удаление одной буквы	5

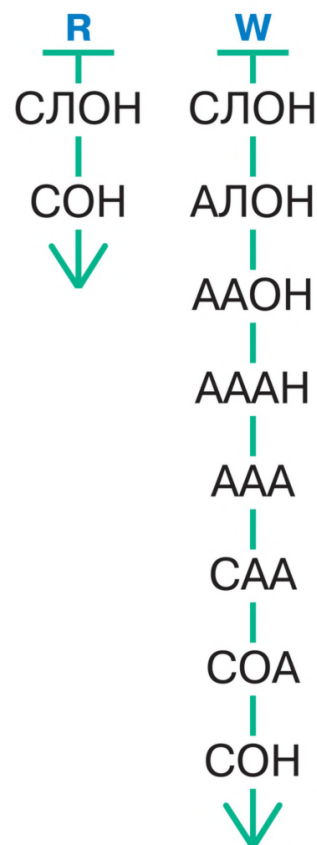




Diagram illustrating the reduction of a word into a root:

СЛОНО (5) → СЛОН (3) → АЛОН (3) → ААОН (1) → АААН (5) → ААА (3) → САА (1) → СОА (3) → СОН

1  
AAAH  
5  
AAA  
3  
CAA  
1  
COA  
3  
COH  
↓



ΠΑΠ-ΚΑ  
ΠΑΠΑΧΑ

↓

ΠΑΠΚΑ  
↓  
ΠΑΠΑΚΑ  
↓  
ΠΑΠΑΧΑ  
↓

Τ  
 ΠΑΠΚΑ  
 —  
 ΠΑΠΑΚΑ  
 —  
 ΠΑΠΑΧΑ  
 ↓

и базы данных, содержащие последовательности известных белков, доступны всем через Интернет.

Выравнивание позволяет предположить, какая последовательность остатков была у общего предка двух видов живых существ. Вот, например, выравнивание двух фрагментов белков: один из них — белок человека, а другой — белок мартышки:

VLSPADK**T**NVKAAWGKVG**A**HAG**E**YGAEALERMFLS  
VLSPADK**S**NVKAAWGKVG**S**HAG**D**YGAEALERMFLS

Значит, можно предположить, что наш общий предок имел такую белковую последовательность:

VLSPADK + NVKAAWGKVG + HAG + YGAEALERMFLS

(плюсами помечены места, буквы в которых нам неизвестны: Т или S, А или S, Е или D).

226

Построй какое-нибудь (необязательно самое дешёвое) превращение слова ТАНКИСТ в слово ТУРИСТ. Напиши последовательность своего превращения и вычисли его стоимость.

227

Вот последовательности G и Q превращений слова ОБЪЕКТ в слово ОБЛИК. Вычисли стоимость каждого превращения и укажи более дешёвое.

**G**  
—  
ОБЪЕКТ  
|  
ОБЕКТ  
|  
ОБЛКТ  
|  
ОБЛИТ  
|  
ОБЛИК  
↓

**Q**  
—  
ОБЪЕКТ  
|  
ОБЛЕКТ  
|  
ОБЛИКТ  
|  
ОБЛИК  
↓

228

Построй какое-нибудь превращение (необязательно самое дешёвое) слова ВРАЗБРОС в слово ВРАЗРЕЗ. Напиши последовательность этого превращения и вычисли его стоимость.



229

Построй превращение слова КОНТАКТ в слово КОМПЛЕКТ, которое стоит 10 баллов.

КОНТА – КТ  
КОМПЛЕКТ

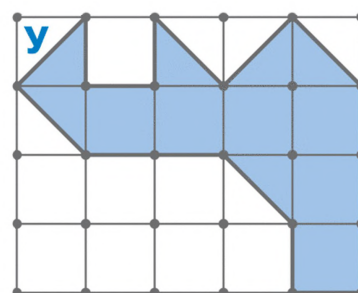


Чтобы построить превращение с заданной стоимостью, удобно сначала *выровнять* заданные слова: расположить одно слово под другим так, чтобы одинаковые буквы (конечно, если они есть) стояли друг под другом. При необходимости можно раздвигать буквы, вставляя чёрточку.

Как именно нужно расположить остальные (неодинаковые) пары букв, зависит от того, какова заданная стоимость. Например, указанное выравнивание предполагает замену Н на М (2 балла), замену Т на П (2 балла), замену А на Л (3 балла) и вставку Е (5 баллов) — всего 12 баллов. Это не та стоимость, какая требуется в задаче. Значит, надо попытаться поставить чёрточку в другом месте и снова подсчитать баллы.

230

Нарисуй, как разрезать многоугольник У, чтобы получились два одинаковых многоугольника на сетке.



231

За какое наименьшее число вопросов можно наверняка отгадать натуральное число, меньшее 1000, если на вопросы можно отвечать только «да» и «нет»?

232

Построй превращение слова СЪЕДОБНЫЙ в слово ОГРОМНЫЙ, которое стоит 15 баллов.

233

Построй превращение слова ПОЛЬСКИЙ в слово КОНСКИЙ, которое стоит меньше 10 баллов.

234

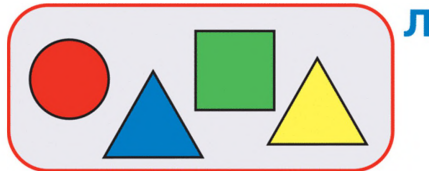
Построй превращение слова ПОДОСИНОВИК в слово ПОДБЕРЁЗОВИК, которое стоит 11 баллов.

Подумай, существует ли более дешёвое превращение слова ПОДОСИНОВИК в слово ПОДБЕРЁЗОВИК. Объясни свой ответ.

# Повторение

235

Построй все подмножества множества Л.



236

Запиши последовательность чисел длины 5, для которой истинны следующие утверждения:

Первый член последовательности — число 3.

Второй член последовательности — число 7.

Каждый член последовательности, кроме первого и последнего, равен среднему арифметическому предыдущего и следующего членов.

237

Даны названия месяцев года на разных языках: армянском, грузинском, современном греческом и еврейском (иврите). Пользуясь другими задачами в этом учебнике, рассортируй слова по языкам. Конечно, сами слова переписывать не надо, достаточно записать номер слова.

1 հիւնիս

2 Σεπτεμβρης

3 ლი

4 Απριλης

5 դեկտեմբեր

6 ივლისი

7 רצבאיטצס

8 փալիս

9 Μαρτης

10 տառ

11 თებერვალი

12 հիւնվար

13 Φλεβαρης

14 სექტემბერი



238

Элементами дерева  $A$  являются буквы, а все последовательности дерева  $A$  выписаны справа. Построй дерево  $A$ , если известно, что следующие утверждения истинны:

В дереве  $A$  всего один элемент первого уровня.

В дереве  $A$  всего 23 элемента.

ДЕРЕВО  
ДРУГ  
ДРОЗД  
ДЕРЕВНЯ  
ДРОВА  
ДЕРЖИ  
ДРОВНИ  
ДЕРЖАВА

239

Расположи все чётные двузначные числа, которые делятся на 7, в порядке убывания.

240

Выдели подмножество множества  $D$ , для которого все следующие утверждения истинны:

Все элементы этого подмножества — двузначные числа.

Каждое число из этого подмножества больше 62.

В этом подмножестве всего 5 элементов.

В этом подмножестве нет чётных чисел.

Каждое число в этом подмножестве делится на 3.

В этом подмножестве нет чисел, в записи которых есть цифра 5.

85 94 78 123 102  
73 97 77 75 67  
90 66 95 68 83  
72 27 69 87 92  
65 117 51 24 89  
57 9 111 63 70  
84 81 88 71 93

D

241

Реши задачу.

Из девяти монет одна фальшивая — более лёгкая. Как найти её двумя взвешиваниями на чашечных весах без гирь? А сколько потребуется взвешиваний для поиска одной монеты из двадцати семи?



242

Даны правила игры *Ромашка*.

### **Правила игры *Ромашка***

Начальная позиция. Ромашка с лепестками (количество лепестков у ромашки определяется дополнительными правилами).

Возможные ходы. На каждом ходу игрок отрывает либо один лепесток, либо два соседних (растущих рядом).

Как определить победителя. Игра заканчивается, если очередной ход сделать невозможно — все лепестки оторваны. Выигрывает тот, кто сделал последний ход.

Исследуй игру *Ромашка* для различных начальных позиций. У кого из игроков есть равновесная выигрышная стратегия? Сформулируй эту стратегию.



Нужно рассмотреть два случая:

- а) если число лепестков в начальной позиции чётно;
- б) если число лепестков в начальной позиции нечётно.

243

Построй дерево И, для которого все следующие утверждения истинны:

Высота дерева И равна четырём.

Все последовательности дерева И — разные.

На каждом уровне дерева И ровно три листа.

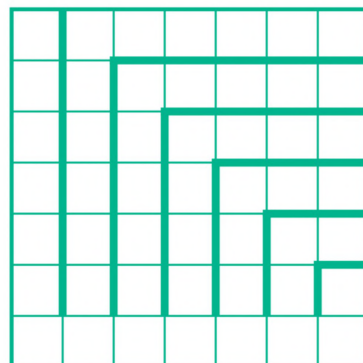
На втором уровне дерева И ровно четыре элемента.

Все элементы дерева И — однозначные числа, которые делятся на 3.



244

Составь такой алгоритм для *Робота*, чтобы, стартовав из любой клетки любого из этих полей, после выполнения твоего алгоритма *Робот* оказался в верхней левой клетке поля.



245

Реши задачу, построив схему с пересечением трёх множеств, как в задаче 149.

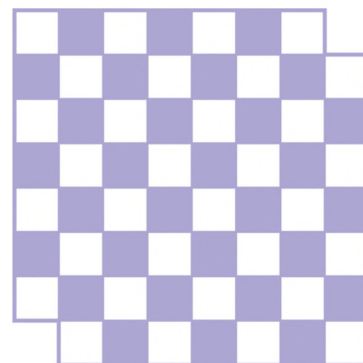

В школьной олимпиаде по математике участвовало 100 человек, по физике — 50, по информатике — 48. Когда учеников спросили, в скольких олимпиадах они участвовали, ответ «В двух» дали вдвое меньше человек, чем ответ «В одной», а ответ «В трёх» — втрое меньше, чем ответ «В одной». Сколько всего учеников участвовало в этих олимпиадах?

246

Робот находится в левом верхнем углу прямоугольного поля. Внутри поля имеется одна горизонтальная стена с одним проходом, идущая от левой до правой стены прямоугольника (проход не прилегает ни к левой, ни к правой стене прямоугольника). Составь алгоритм, при выполнении которого Робот переместится в правый нижний угол прямоугольника.

247

Решите задачу, используя метод половинного деления. Имеется стопка из восьми монет, одна из которых фальшивая (она отличается по весу от всех остальных). Как, имея четыре нефальшивые монеты и чашечные весы, найти фальшивую монету за три взвешивания?



248

Можно ли разбить на «костяшки домино» (каждая из двух клеток) шахматную доску без противоположных углов — полей  $a1$  и  $h8$ ?



249

На окружности нарисованы 20 точек. Двое игроков по очереди соединяют отрезком любые две из этих точек так, чтобы никакие два отрезка не пересекались (но два отрезка могут иметь общий конец). Проигрывает тот, кто не может сделать следующий ход. У кого из игроков есть выигрышная стратегия? Опиши эту стратегию.

250

Робот находится в левом верхнем углу прямоугольного поля шириной 9 и высотой 7 клеток. Нарисуй такое поле и стенки на нём, чтобы, выполнив алгоритм спуск, Робот оказался в правом нижнем углу поля.

```

алг спуск
  дано |
  надо |
нач
  нц 8 раз
    если снизу свободно
      то вниз
    иначе вправо
  все
кц
  нц 3 раз
    если снизу свободно
      то вниз
    все
  кц
  нц 3 раз
    если справа свободно
      то вправо
    все
  кц
кон

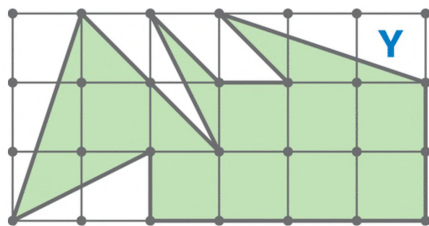
```

251

Сосчитай, сколькими способами 7 человек могут встать в очередь к театральной кассе.

252

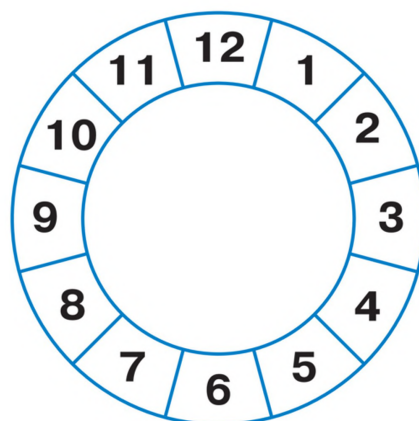
Нарисуй, как разрезать многоугольник Y, чтобы получились два многоугольника на сетке одинаковой площади.





**253**

Двое игроков по очереди переводят часовую стрелку на два или на три часа вперёд. В начальной позиции часовая стрелка указывает на 12. Побеждает тот, кто первым поставит стрелку на 6 (прежде чем остановиться на цифре 6, стрелка может сделать несколько оборотов). У кого из игроков есть выигрышная стратегия?



Для решения задачи используй специальную круглую числовую линейку, которую можно вырезать из вкладыша тетради проектов. Обрати внимание: после того как ты обойдёшь круглую линейку один раз, некоторые позиции останутся незакрашенными. Чтобы раскрасить все позиции, нужно будет обойти линейку несколько раз.

**254**

Прочитай рецепт приготовления торта «Пальчики оближешь». Изобрази процесс приготовления торта в виде дерева, где листья — необходимые продукты, элемент первого уровня — готовый торт.

Приготовление коржей. Сначала взбить три яйца с одним стаканом сахарного песка до получения густой однородной пены. Затем добавить полпачки маргарина и один стакан муки. Всё хорошо перемешать и испечь два коржа.

Приготовление крема. Сварить густую манную кашу из трёх с половиной столовых ложек манки и двух стаканов молока. Затем остудить, добавить одну пачку сливочного масла и один стакан сахарного песка, хорошо перемешать.

Приготовление глазури. Растопить одну шоколадку в небольшом количестве молока.

Приготовление торта. Выложить на один из коржей весь крем, накрыть другим коржом и полить сверху глазурью. Поставить на несколько часов в холодильник — и торт готов.

**255**

Коля и Петя написали своему другу Васе записки, в каждой из которых зашифровано сообщение. Мальчики шифровали сообщение следующим образом: слова сообщения спрятаны в тексте, не связанном по содержанию со смыслом сообщения. Слова сообщения в каждом из текстов спрятаны без определённого правила, но порядок слов сообщения при этом не изменяется.

Прочитай записки двух мальчиков. В каждой из них зашифровано одно и то же сообщение, состоящее из двух предложений. Запиши текст зашифрованного сообщения. Поставь знаки препинания.

### **Записка Коли.**

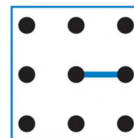
Сегодня состоялся сбор представителей классов по поводу турслёта, который будет завтра. В девять утра началось обсуждение. Оно могло затянуться до вечера, у меня даже разболелась голова. Вышел из школы и встретил старого друга, мы оба очень обрадовались. Он нёс кораблик из дерева. Он сказал: «Возьми на память». Я подарил ему карту нашего района, есть там и наш дом, а также подробный план прилегающих дворов.

### **Записка Пети.**

На биологии рассказывали про сбор хлопка. На завтра задали найти ответы в учебнике на девять вопросов. На перемене обсуждали программу предстоящего вечера. Вообще-то у нас уже был план, но его переделали так, что не осталось ни одного старого конкурса. Стенгазету хотели сделать в виде дерева. Вдруг Колька возьми да и скажи: «Давайте сделаем карту школы, пометим всё самое интересное. Главное — не забыть директора и наш класс». Все начали спорить, и план вечера так и остался незаконченным.

**256**

Найди выигрышную стратегию для Второго в игре *Ползунок* на поле  $3 \times 3$  при условии, что Первый на первом ходу соединит центральную точку поля с одной из боковых.

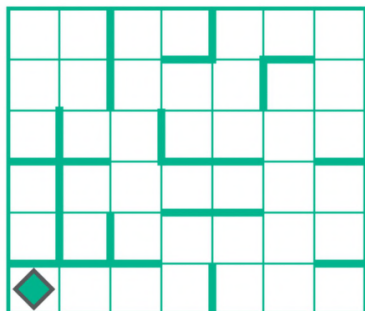




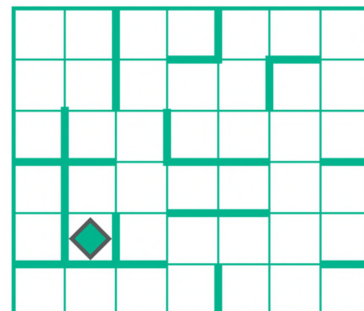
**257**

Составь такой алгоритм, чтобы после его выполнения из каждого из данных начальных состояний *Робот* оказался в какой-нибудь клетке верхнего ряда поля.

Начальное состояние 1:



Начальное состояние 2:

**258**

Реши задачу, используя метод половинного деления.

Имеется 68 алмазов, все алмазы разные по весу. Как за 100 взвешиваний на чашечных весах найти самый лёгкий и самый тяжёлый алмаз?

**259**

Реши задачу.

Все трое внуков Прохора Кольцова — Иван, Степан и Никифор — чем-то на него похожи (цветом глаз, цветом волос или формой носа), но ни цветом глаз, ни цветом волос, ни формой носа не похожи друг на друга. Каждые две дочери каждого из внуков Прохора похожи между собой именно тем, чем они похожи на отца, а все три его дочери — тем, чем их отец похож на своего деда Прохора. Вот описание дочерей Ивана, Степана и Никифора. Опиши деда Прохора и его внуков.

#### Дочери Ивана

Настя: глаза карие, волосы русые, нос с горбинкой.

Ася: глаза голубые, волосы каштановые, нос с горбинкой.

Даша: глаза карие, волосы каштановые, нос с горбинкой.

#### Дочери Степана

Галя: глаза зелёные, волосы рыжие, нос вздёрнутый.

Валя: глаза карие, волосы рыжие, нос с горбинкой.

Люба: глаза зелёные, волосы рыжие, нос вздёрнутый.

#### Дочери Никифора

Оля: глаза голубые, волосы русые, нос прямой.

Поля: глаза голубые, волосы каштановые, нос прямой.

Лена: глаза голубые, волосы русые, нос вздёрнутый.

260

Пользуясь решением задачи 259, определи истинностное значение утверждений, заполни таблицу истинности 1. Используя таблицу 1, заполни таблицу 2.

Таблица 1

Имя	Утверждение	Значение
<b>A</b>	У Ивана карие глаза.	
<b>D</b>	У Степана нос с горбинкой.	
<b>F</b>	У Никиты русые волосы.	
<b>L</b>	У Прохора голубые глаза.	

Таблица 2

Утверждение	Утверждение (полностью)	Значение
A и D		
D и L		
L или F		
D или A		
не F		
не D		

261

Даны правила игры *Три кучи камешков*. Известно, что в игре *Три кучи камешков* Первый имеет равновесную выигрышную стратегию. Сформулируй эту стратегию.

### **Правила игры *Три кучи камешков***

Начальная позиция. Три кучи камешков, во всех кучах камней поровну (сколько именно, устанавливается дополнительными правилами).

Возможные ходы. На каждом ходу игрок может взять любое число камешков, но только из одной кучи.

Как определить победителя. Игра заканчивается, если все камешки закончились. Выигрывает игрок, который забрал последний камешек.



262

Выпиши все числа, меньшие 600, большие 400 и такие, что число десятков в каждом из них меньше числа сотен и в каждом есть две одинаковые цифры.

263

Разведчики прячут слова донесения в тексте. Один разведчик размещает слова в тексте в том же порядке, как они идут в донесении, а другой размещает слова не по порядку. В каждой шифровке содержится одно и то же донесение.



Запиши текст этого донесения. Поставь знаки препинания. В текстах-шифровках попадают случайно совпадающие (лишние) предлоги. Выбери нужные предлоги, учитывая падежи существительных.



### Шифровка первого разведчика.

На собрание явка обязательна для всех проживающих в доме. Будут обсуждаться следующие вопросы:

1. У жильцов верхних этажей низкая температура горячей воды.
2. Крыша сараев провалилась.
3. Новое решение домоуправления предписывает выделить место под детскую площадку.
4. Квартира под музыкальной школой отапливается плохо.
5. Необходимо выделить людей для работы с цветочным палисадником, находящимся рядом с магазином.

### Шифровка второго разведчика.

Затея с субботником провалилась. Явка оказалась низкой. Ни у кого в доме не нашлось ни лопаты, ни лейки для воды. Пришлось связываться с магазином. Выбрали новое место для клумбы, но, пока ходили за цветочным саженцем, все под разными предлогами разошлись. Видимо, своя квартира оказалась милее общего двора.

264

Запиши шифровку, длина которой не больше 18 латинских букв, такую, чтобы в ней содержались 2 слова — БЕЛЫЙ и НОГУ. Для того чтобы выполнить задание, тебе понадобится использовать наложение шифровок, подобное тому, о котором говорится в задаче 218.

265

В таблице размером  $7 \times 7$  клеток двое игроков по очереди закрашивают клетки так, чтобы раскрашенные клетки не имели общих сторон. Проигрывает тот, кто не может сделать ход. У кого из игроков есть выигрышная стратегия в этой игре? Опиши эту стратегию.

266

Выпиши множество А всех чисел, меньших 30, которые делятся на 2. Выпиши множество В всех чисел, меньших 30, которые делятся на 3. Найди пересечение и объединение множеств А и В.

267

Роботу разрешается выполнять только алгоритм **УГОЛОК** (см. справа), а больше никакие команды движения выполнять нельзя. Робот стоит на поле, где-то внизу под ним — горизонтальная граница поля, а справа — вертикальная граница поля.

**алг** УГОЛОК

**нач**

вправо

вниз

**кон**

Внутренних стен на поле нет. Используя алгоритм **УГОЛОК** как вспомогательный, построй алгоритм, выполняя который Робот дойдёт до одной из этих стен.

268

Реши задачу.

Из трёх одинаковых по виду колец одно несколько легче, чем каждое из двух других (а два других весят одинаково). Как одним взвешиванием найти более лёгкое кольцо?

269

На рисунке справа показано положение на поле клеток с именами А, В, С, D и Е. Можно ли на этом поле расставить стенки так, чтобы условия имели истинностные значения, указанные в таблице?

		А		
		В	С	
		Е	Д	

а)

	А	В	С	Д	Е
сверху свободно	И	Л	И	Л	И
справа свободно	Л	Л	Л	Л	Л
снизу свободно	И	Л	И	И	И
слева свободно	Л	Л	И	Л	И

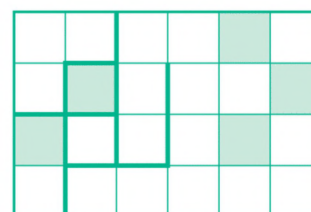


б)

	А	В	С	Д	Е
сверху свободно	И	Л	И	И	Л
справа свободно	Л	Л	Л	Л	Л
снизу свободно	Л	Л	И	И	И
слева свободно	Л	Л	Л	Л	И

270

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.



	А	В	С	Д	Е	Ф
слева свободно и справа свободно	Л	Л	Л	И	Л	И
не сверху свободно и не клетка закрашена	Л	И	Л	Л	И	И
снизу свободно или не слева свободно	И	И	Л	И	Л	Л

271

Реши задачу, используя метод половинного деления. Из четырёх деталей одна отличается по весу от остальных. Можно ли выделить её двумя взвешиваниями на чашечных весах без гирь?

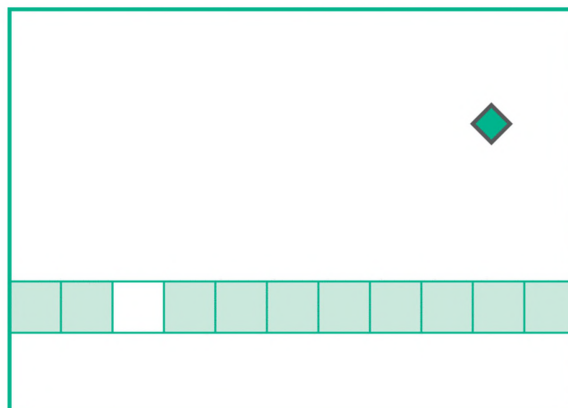
272

Реши задачу.  
В бутылке, стакане, кувшине и банке находятся молоко, лимонад, квас и вода. Известно, что в бутылке не вода и не молоко, кувшин стоит рядом с лимонадом и далеко от кваса, в банке не лимонад и не вода. Стакан стоит рядом с банкой и далеко от молока. В какой сосуд налита каждая жидкость?  
Для решения задачи удобно построить схему, выписав в одном столбце названия всех сосудов, а в другом — названия всех напитков.



273

Робот находится в прямоугольнике, в котором нет внутренних стен, закрасены только клетки одного ряда — все, кроме какой-то одной. Построй алгоритм, выполнив который Робот остановится именно в этой единственной незакрашенной клетке этого ряда.



274

Реши лингвистическую задачу. Вот несколько айнских числительных (в латинской транскрипции):

3 — re

11 — sine ikasma wan

22 — tu ikasma hot

37 — arwan ikasma wan e tu hot

47 — arwan ikasma tu hot

93 — re ikasma wan e asikne hot

135 — asikne ikasma wan e arwan hot

Определи, какое число по-айнски записывается так:

wan e re hot.

Запиши по-айнски числа: 1, 5, 12, 53, 67, 85, 100.

275

Реши задачу.

Вася и Петя играют в игру. Вася записывает в каждой строке таблицы  $4 \times 4$  клетки последовательность из букв М и А так, чтобы последовательности во всех строках были разные. Затем Вася закрывает каждую клетку белым квадратиком и передаёт Пете. Петя должен открыть любые четыре клетки таблицы и написать последовательность длины 4, составленную из М и А, которой в таблице нет. Если Пете это удастся, то он выиграл, если нет — выиграл Вася. Какие четыре клетки Петя должен открыть, чтобы наверняка выиграть?



# Компьютерный проект

## С видеокамерой в руках...

Дорогие ребята!

Этот проект посвящён видеосъёмке. Сейчас трудно найти человека, который не делал бы видеозаписи. Функция видеосъёмки имеется на большинстве моделей мобильных телефонов и коммуникаторов. Владельцы планшетных компьютеров постоянно используют видеозапись для фиксации интересных событий и быстро делятся этой информацией с другими, размещая её в Интернете. Мы постараемся не просто зафиксировать увиденное глазом, а создать интересный ресурс, используя современные возможности цифровой видеосъёмки и монтажа.

Тема вашей работы может быть любой, главное, чтобы она была вам понятна и интересна.

В качестве примера мы рассмотрим технологию работы над фильмом об экскурсии. Идея проекта — сделать фильм, который расскажет зрителям о каком-нибудь интересном месте, в котором вам и вашим одноклассникам удалось побывать вместе, и передаст впечатления от увиденного. В нашей стране большое количество памятников истории и культуры. Надеемся, что в этом учебном году вы уже посетили один из них или поедете на экскурсию в самое ближайшее время. Не забудьте взять видеокамеру на штативе — сделанные во время экскурсии видеозаписи будут использованы в нашем фильме. Имея даже одну камеру, вы сможете снимать, передавая её друг другу. Накануне экскурсии посмотрите подсказки начинающим видеолюбителям, которые имеются в описании этого проекта в тетради проектов. Постарайтесь не забыть о них во время съёмки.

Вы уже имеете опыт по созданию мультфильма, поэтому организация работы над фильмом не станет для вас неожиданностью. Готовясь к уроку, постарайтесь ответить на следующие вопросы:

- О чём ваш фильм?
- Для кого ваш фильм?
- Сколько приблизительно времени будет длиться фильм?
- Какие сюжеты будут в вашем фильме?

На все эти вопросы можно ответить при составлении сценарного плана. Создайте свою творческую группу из 3—5 че-



ловек, соберитесь вместе и посмотрите фотоматериалы по теме будущей экскурсии в книгах или в Интернете, продумайте план фильма (см. монтажный лист в тетради проектов). Хорошая подготовка к экскурсии позволит не растеряться, получив в руки видеокамеру, и быстро найти взглядом интересные «картинки» для вашего фильма. Вы можете включить в ваш сценарий интервью с участниками экскурсии, отснять которое можно будет на уроке уже после экскурсии.

Чтобы превратить видеозаписи в короткий фильм, нам нужно будет поработать с ними два урока. На первом уроке надо будет перенести отснятый видеоматериал в компьютерную программу для редактирования видео. Кроме того, нужно сделать всю основную работу по подбору и редактированию клипов. Добавление титров и работу со звуком можно отложить на второй урок.

В работе над фильмом нам поможет опыт мультипликации, полученный в 5 классе, где мы использовали функцию фото. В этом проекте мы используем функцию видеосъёмки. В результате при переносе информации на компьютер мы получаем цепочку, состоящую из видеоклипов, которые можно просматривать, удалять из них лишние фрагменты и т. п.

Для мультфильма	Для фильма
1. Съёмка, создание видеоцепочки.	1. Съёмка, создание видеоцепочки.
2. Расстановка кадров и установка длительности.	2. Импорт видеоматериала.
3. Добавление титров.	3. Просмотр и расстановка клипов на «монтажной линейке», редактирование клипов.
4. Озвучивание (запись голоса).	4. Добавление титров.
5. Музыкальное сопровождение	5. Озвучивание (запись голоса).
	6. Музыкальное сопровождение

Этапы создания мультфильма и видеофильма различаются лишь одним этапом. Именно эти этапы нам и предстоит пройти за два урока.

Постарайтесь справиться с работой быстрее, потому что оставшееся время можно будет потратить на просмотр и обсуждение ваших удивительных фильмов. Не забудьте обсудить с учителем, как можно посмотреть фильм со всей семьей, ведь вашим родителям это тоже будет интересно.



# Содержание курса

Введение.....	3
Сортировка: упорядочение и классификация.....	4
Дерево сортировки .....	10
Словари .....	15
Проект. Словари .....	тетрадь проектов
Проект. Сортировки .....	тетрадь проектов
Исполнители и алгоритмы .....	22
Практикум. Исполнители .....	компьютер
Вспомогательный алгоритм.....	25
Практикум. Вспомогательный алгоритм.....	компьютер
Дерево перебора вариантов. Дерево перебора подмножеств .....	31
Поиск кратчайшего пути .....	38
Алгоритмы: цикл «N раз» .....	43
Практикум. Цикл «N раз».....	компьютер
Повторение .....	50
Игры с полной информацией .....	55
Дерево игры.....	57
Команды-запросы Робота. Условие .....	61
Практикум. Команды-запросы Робота.....	компьютер
Выигрышная стратегия.....	72
Выигрышные и проигрышные позиции.....	73
Выигрышные стратегии.....	80
Цикл «пока» .....	87
Свойства цикла «пока» .....	89
Составление алгоритма с циклом «пока» .....	91
Практикум. Цикл «пока».....	компьютер
Равновесные выигрышные стратегии .....	96
Составные условия: слова «и», «или», «не».....	106
Практикум. Составные условия .....	компьютер
Биоинформатика. Белки и ДНК. Почему дети похожи на родителей?.....	114
Шифрование.....	117

Биоинформатика. Как кодируются белки .....	<b>122</b>
Автомат-сортировщик.....	<b>128</b>
Метод половинного деления.....	<b>132</b>
Биоинформатика. Как изучают белки .....	<b>135</b>
Биоинформатика. Сравнение белков.....	<b>138</b>
Превращение слов .....	<b>139</b>
Повторение .....	<b>143</b>
Компьютерный проект. С видеокамерой в руках ... (практика работы с аудио- и видеоматериалами).....тетрадь проектов, компьютер	<b>156</b>

## Ответы к задачам:

№ 13: 10,5 ед. кв. № 20: 6 слов. № 51: 13,5 ед. кв.  
 № 62: 24 путями. № 68: цифра 4. № 79: 20 брёвен. № 87: 59 мин.  
 № 91: 15 человек. № 115: 24 способами. № 120: 10 руко-  
 пожатий. № 127: 10 путями. № 149: 9 учеников, 1 ученик.  
 № 162: 12 чисел. № 201: 90 000 000 нуклеотидов. № 204: А и Г.  
 № 207: Саня и Даня. № 208: 63 типа. № 225: 24 числа.  
 № 231: за 10 вопросов. № 248: нельзя.



Учебное издание

**Семёнов Алексей Львович**  
**Рудченко Татьяна Александровна**

## **ИНФОРМАТИКА**

**6 класс**

**Учебник**

Центр развития углублённого и профильного образования, функциональной грамотности, технологии и ИКТ-компетенций

Ответственный за выпуск *Е. А. Баклашова*

Редакторы *О. В. Платонова, Е. С. Карауш*

Художественный редактор *Т. В. Глушкова*

Компьютерная графика *Н. А. Артемьевой*

Технический редактор и верстальщик *Н. В. Лукина*

Корректор *Н. В. Игошева*

Подписано в печать 27.01.2022. Формат 70×90/16. Гарнитура SchoolBookCSanPin.

Усл. печ. л. 11,7. Уч.-изд. л. 8,35. Тираж экз. Заказ № .

Акционерное общество «Издательство «Просвещение».

Российская Федерация, 127473, г. Москва, ул. Краснопролетарская, д. 16, стр. 3,  
этаж 4, помещение I.

Адрес электронной почты «Горячей линии» — **[vopros@prosv.ru](mailto:vopros@prosv.ru)**.

## Обратная таблица генетического кода

Кодон	Остаток	Буква	Кодон	Остаток	Буква
AAA	Лизин	K	CAA	Глутамин	Q
AAG	Лизин	K	CAG	Глутамин	Q
AAC	Аспарагин	N	CAC	Гистидин	H
AAT	Аспарагин	N	CAT	Гистидин	H
AGA	Аргинин	R	CGA	Аргинин	R
AGG	Аргинин	R	CGG	Аргинин	R
AGC	Серин	S	CGC	Аргинин	R
AGT	Серин	S	CGT	Аргинин	R
ACA	Треонин	T	CCA	Пролин	P
ACG	Треонин	T	CCG	Пролин	P
ACC	Треонин	T	CCC	Пролин	P
ACT	Треонин	T	CCT	Пролин	P
ATA	Изолейцин	I	CTA	Лейцин	L
ATG	Метионин	M	CTG	Лейцин	L
ATC	Изолейцин	I	CTC	Лейцин	L
ATT	Изолейцин	I	CTT	Лейцин	L
GAA	Глутамат	E	TAA	Стоп-кодон	*
GAG	Глутамат	E	TAG	Стоп-кодон	*
GAC	Аспартат	D	TAC	Тирозин	Y
GAT	Аспартат	D	TAT	Тирозин	Y
GGA	Глицин	G	TGA	Стоп-кодон	*
GGG	Глицин	G	TGG	Триптофан	W
GGC	Глицин	G	TGC	Цистеин	C
GGT	Глицин	G	TGT	Цистеин	C
GCA	Аланин	A	TCA	Серин	S
GCG	Аланин	A	TCG	Серин	S
GCC	Аланин	A	TCC	Серин	S
GCT	Аланин	A	TCT	Серин	S
GTA	Валин	V	TTA	Лейцин	L
GTG	Валин	V	TTG	Лейцин	L
GTC	Валин	V	TTC	Фениланин	F
GTT	Валин	V	TTT	Фениланин	F

Приведены все 64 кодона в «ДНК-алфавитном» порядке, т. е. порядок букв в алфавите такой: А, G, C, Т.

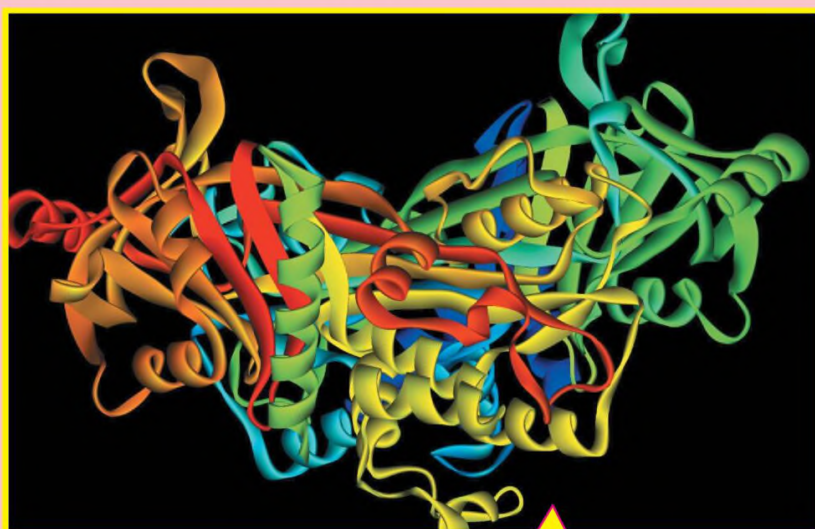
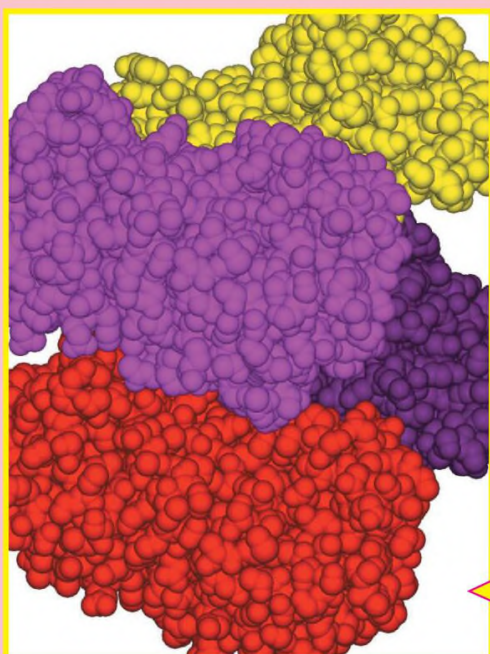
В таблице генетического кода, которую вы будете изучать в старших классах, вместо буквы Т стоит U. Это связано с тем, что создание белков в организме происходит в несколько этапов, при этом создаётся промежуточное звено – молекула РНК, в которой тимин (Т) заменяется урацилом (U).



## Таблица генетического кода

Остаток	Буква	Кодоны					
Аланин	A	GCT	GCC	GCA	GCG		
Цистеин	C	TGT	TGC				
Аспартат	D	GAT	GAC				
Глутамат	E	GAA	GAG				
Фениланин	F	TTT	TTC				
Изолейцин	I	ATA	ATC	ATT			
Лизин	K	AAA	AAG				
Лейцин	L	CTT	CTC	CTA	CTG	TTA	TTG
Метионин	M	ATG					
Аспарагин	N	AAT	AAC				
Пролин	P	CCT	CCC	CCA	CCG		
Глутамин	Q	CAA	CAG				
Аргинин	R	CGT	CGC	CGA	CGG	AGA	AGG
Серин	S	TCT	TCC	TCA	TCG	AGT	AGC
Треонин	T	ACT	ACC	ACA	ACG		
Валин	V	GTT	GTC	GTA	GTG		
Триптофан	W	TGG					
Тирозин	Y	TAT	TAC				
Стоп-кодон	*	TAA	TAG	TGA			

Слово «белок» происходит от названия белка куриного яйца. В курином белке есть молекулы разных видов белков. Больше всего молекул белка под названием *овальбумин* (от латинских слов *ovum* – Яйцо и *albus* – белый). Молекула овальбумина состоит примерно из 1500 звеньев. В живых организмах цепочки многих белков сложены в плотные структуры.



Поверхность  
молекулы  
овальбумина

Ход цепи  
молекулы  
овальбумина