



Федеральная служба по надзору в сфере образования  
и науки

ФГБНУ «Федеральный институт педагогических  
измерений»

**МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ**  
**обучающимся**  
**по организации индивидуальной**  
**подготовки к ОГЭ 2026 года**

**ИНФОРМАТИКА**

Москва, 2026

Авторы-составители: С.С. Крылов, В.Р. Лещинер

Методические рекомендации предназначены для обучающихся 9 класса, планирующих сдавать ОГЭ по информатике, содержат советы разработчиков контрольных измерительных материалов ОГЭ и полезную информацию для организации индивидуальной подготовки к ОГЭ. В пособии описаны структура и содержание контрольных измерительных материалов ОГЭ, приведён индивидуальный план подготовки к экзамену, указаны темы, на освоение/повторение которых целесообразно обратить особое внимание. Даны рекомендации по выполнению разных типов заданий, работе с банком заданий ОГЭ и другими дополнительными материалами, указаны полезные ссылки на информационные материалы ФИПИ.

### Дорогие друзья!

Скоро вам предстоит сдать основной государственный экзамен (ОГЭ) по информатике. Ваша основная задача – показать хорошую подготовку по информатике. Данные рекомендации помогут вам в подготовке к экзамену.

В 2026 г., как и в предыдущие годы, вариант КИМ ОГЭ по информатике состоит из двух частей, различающихся типом ответа на предложенные задания: в части 1 собраны задания с кратким ответом, в части 2 – с развёрнутым ответом. Задания каждой части расположены по возрастанию сложности, поэтому задания по одним и тем же разделам курса информатики, различающиеся уровнем сложности, в КИМ могут находиться не рядом друг с другом. В 2026 г., как и в 2025 г., работа ОГЭ включает в себя 16 заданий. Задания 11–16 выполняются на компьютере. Задание 13 альтернативное: нужно выполнить либо задание 13.1, либо задание 13.2.

В таблице 1 представлено распределение заданий по разделам курса информатики.

*Таблица 1*

Номер тематического блока	Название тематического блока	Номер задания	Какое умение проверяется
1	Цифровая грамотность	11	Осуществлять поиск информации в файлах и каталогах компьютера
		12	Определять количество и информационный объём файлов, отобранных по некоторому условию
		7	Знать принципы адресации в сети Интернет
		8	Понимать принципы поиска информации в сети Интернет
2	Теоретические основы информатики	1	Оценивать объём памяти, необходимый для хранения текстовых данных
		2	Декодировать кодовую последовательность
		3	Определять истинность составного высказывания
		10	Записывать числа в различных системах счисления
		4	Анализировать простейшие модели объектов
		9	Анализировать информацию, представленную в виде схемы
3	Алгоритмы и программирование	5	Анализировать простые алгоритмы для конкретного исполнителя с фиксированным набором команд
		6	Формально исполнять алгоритмы, записанные на языке программирования
		15	Создавать и выполнять программы для заданного исполнителя
		16	Создавать и выполнять программы на универсальном языке программирования
4	Информационные технологии	13	Создавать презентации (вариант задания 13.1) или текстовый документ (вариант задания 13.2)
		14	Проводить обработку большого массива данных с использованием средств электронной таблицы

На основании таблицы 1 можно составить индивидуальный план подготовки к экзамену, учитывая ваш текущий уровень освоения учебного материала. Рекомендуем вам ознакомиться с демонстрационным вариантом КИМ этого года, попробовать выполнить тренировочный вариант ОГЭ, например демонстрационный вариант или вариант, самостоятельно собранный из заданий открытого банка заданий ОГЭ, размещённого на официальном сайте ФГБНУ «ФИПИ» <[www.fipi.ru](http://www.fipi.ru)>. Также можно воспользоваться различными сборниками вариантов, при этом следует обращать внимание на соответствие структуры и тематики заданий этих вариантов официальному демонстрационному варианту КИМ. Если выполнение заданий по какой-либо теме вызвало затруднения, то на изучение и повторение соответствующей темы следует запланировать больше времени. Не следует изначально планировать себе слишком жёсткий график, который вы вряд ли сможете соблюдать, – будьте реалистом.

В графах «Пройдено» и «Необходимо изучить/повторить» таблицы 2 отметьте не только прохождение соответствующего учебного материала, но и параграфы учебников или других учебных материалов, которые соответствуют нужной теме. Так вы сформируете свой индивидуальный план подготовки к экзамену.

Таблица 2

№	Элементы содержания	Пройдено	Необходимо изучить/повторить	Период времени
<b>1. Цифровая грамотность</b>				
1.1	Программное обеспечение персонального компьютера. Операционная система. Файловая система. Файлы и каталоги. Имя файла, атрибуты. Файловый менеджер. Навигация. Поиск файла			
1.2	Принципы построения глобальной сети Интернет. Сетевые протоколы. Адреса интернет-ресурсов			
1.3	Диаграммы Эйлера – Венна, формула включений и исключений			
<b>2. Теоретические основы информатики</b>				
2.1	Дискретная форма представления информации. Единицы измерения количества информации			
2.2	Кодирование символов одного алфавита с помощью кодовых слов в другом алфавите, кодовая таблица, декодирование. Кодирование текстовой информации			
2.3	Высказывания. Истинность и ложность высказываний. Простые и составные высказывания. Логические значения, операции и выражения			
2.4	Позиционные системы счисления. Запись десятичного числа в системах счисления			

	с основаниями 2, 8, 16			
2.5	Графы. Представление графа в виде схемы и таблицы			
<b>3. Алгоритмы и программирование</b>				
3.1	Основные алгоритмические конструкции: линейная последовательность операторов, цикл, ветвление			
3.2	Синтаксис, типы данных, операции, выражения одного из языков программирования (C++, Школьный алгоритмический язык, Python, Pascal)			
3.3	Программирование для заданного алгоритмического исполнителя			
3.4	Использование среды программирования на одном из языков (C++, Школьный алгоритмический язык, Python, Pascal) для создания простых программ			
<b>4. Информационные технологии</b>				
4.1	Подготовка компьютерных презентаций. Включение в презентацию графических объектов			
4.2	Текстовые документы и их структурные элементы (страница, абзац, строка, слово, символ). Текстовый процессор – инструмент создания, редактирования и форматирования текстов. Включение в текстовый документ списков и таблиц			
4.3	Электронные (динамические) таблицы. Выделение диапазона таблицы и сортировка его элементов, формулы и вычисления по ним, построение графиков и диаграмм			

Рассмотрим подробнее выполнение заданий каждого тематического блока экзаменационной работы и типичные ошибки, допущенные участниками ОГЭ прошлых лет, в таблицах 3–6.

Таблица 3

### 1. Цифровая грамотность

Номер задания	Рекомендации по выполнению	Типичные затруднения, ошибки, рекомендации по их преодолению
11	<p>Необходимо сначала среди специально установленных на компьютере для использования экзаменуемым файлов и каталогов найти требуемый каталог, затем в этом каталоге с помощью встроенных средств поиска операционной системы найти нужный файл по ключевому слову и далее, используя при необходимости поисковые средства текстового редактора, найти ответ на вопрос задания.</p> <p>Файл может быть представлен в различных текстовых форматах; вам следует выбрать формат, соответствующий используемому вами редактору</p>	<p>Причиной невыполнения этого задания может являться недостаточное овладение экзаменуемым базовыми навыками работы с операционной системой персонального компьютера. Следует также внимательно относиться к выбору ключевых слов для поиска</p>
12	<p>Как и при выполнении предыдущего задания, следует сначала найти нужный каталог, затем с помощью файлового менеджера операционной системы отобрать файлы по указанному признаку и записать в ответе их количество</p>	<p>Как и для прошлого задания, источником затруднений при выполнении может являться недостаточная сформированность базовых навыков компьютерной грамотности – умения работать с файловой системой персонального компьютера</p>
7	<p>Перед выполнением этого задания необходимо повторить правила адресации сетевых ресурсов (URL), затем найти среди предложенных фрагментов начало сетевого адреса, содержащее используемый для доступа к файлу протокол. Далее нужно последовательно выбирать фрагменты, соответствующие следующему справа элементу адреса</p>	<p>Обычно ошибки в выполнении этого задания связаны с нетвёрдым знанием правил построения адреса сетевого ресурса</p>
8	<p>Рекомендуется построить соответствующую условию задания диаграмму Эйлера – Венна, затем применить формулу включений и исключений</p>	<p>Типичной ошибкой является неверное применение формулы включений и исключений, в частности неверного понимания, когда нужно использовать операцию объединения, а когда – пересечения множеств</p>

## 2. Теоретические основы информатики

Номер задания	Рекомендации по выполнению	Типичные ошибки
1	<p>Типичным заданием этой линии является задание, приведённое в демонстрационном варианте.</p> <p>Для его выполнения сначала выразите все величины количества информации, встречающиеся в условии, в байтах. Затем определите, сколькими байтами кодируется один символ, обычно это один байт (в восьмиразрядной кодировке текста) или два байта (в шестнадцатиразрядной кодировке). Далее выясните, на сколько символов уменьшилась строка, и выберите из предложенных вариантов ответа слово нужной длины</p>	<p>В этом задании ошибки могут быть допущены при переходе измерения информации от бит к байтам, а также при определении количества байт, занимающего код одного символа. Также следует помнить, что пробелы и знаки препинания тоже являются символами и кодируются тем же количеством байт</p>
2	<p>Это задание можно выполнить, например, следующим образом. Последовательно перебираем коды букв из таблицы, останавливаясь на том коде, который совпадает с началом зашифрованного сообщения. Если подходящий код в таблице единственный, то вместо фрагмента кода в сообщении пишем соответствующую букву и продолжаем алгоритм сначала, пока не расшифруем всё слово.</p> <p>Что делать в ситуации, когда подходят несколько букв из таблицы с кодом разной длины, один из которых является началом другого?</p> <p>В этом случае нужно выбрать один из вариантов, а остальные отложить (записать) и продолжить алгоритм сначала. Если на каком-то этапе окажется, что ни один код не подходит, то придётся рассмотреть отложенные варианты</p>	<p>Как и в большинстве простых заданий, основные ошибки происходят из-за торопливости и невнимательности. Не спешите, тогда сможете успешно выполнить задание.</p>
3	<p>При выполнении заданий, подобных приведённым в демонстрационном варианте, можно использовать рисунок с числовой осью, чтобы наглядно выделить соответствующие условию числа</p>	<p>Типичные ошибки связаны с неверным применением логической операции (конъюнкции вместо дизъюнкции и наоборот), а также с невнимательностью при выборе минимального (максимального) числа из заданного множества. Также следует помнить, что отрицанием условия «больше» является условие «меньше или</p>

		равно» (аналогично для «меньше» – «больше или равно»)
10	<p>В большинстве случаев максимальное (минимальное) число можно найти без перевода чисел из одной системы счисления в другую. Рассмотрим типичное задание: найти максимальное из чисел <math>23_{16}</math>, <math>32_8</math>, <math>11110_2</math> и записать ответ в десятичной системе счисления.</p> <p>Известно, что один восьмеричный разряд соответствует трём двоичным разрядам (триаде), а один шестнадцатеричный разряд – четырём двоичным разрядам (тетраде).</p> <p>Поэтому в числе <math>23_{16}</math> – восемь значащих двоичных разрядов, в числе <math>32_8</math> – шесть, в числе <math>11110_2</math>, очевидно, – пять. Таким образом, максимальное число <math>23_{16} = 2 \times 16 + 3 = 35</math>.</p> <p>Другой способ решения – перевести все числа в десятичную систему и выбрать максимальное</p>	<p>Большинство ошибок при выполнении заданий этой линии происходит из-за неверных действий при переводе из одной системы счисления в другую. Так, например, обучающиеся иногда забывают, что переводить числа из двоичной системы по тетрадам и триадам можно только в восьмеричную и шестнадцатеричную системы соответственно</p>
4	<p>Рекомендуемый способ выполнения этого задания – построить по таблице схему дорог, выписать возможные пути, соединяющие указанные пункты, и определить кратчайший среди них</p>	<p>Иногда экзаменуемый ограничивается первым найденным путём, что может привести к ошибочному результату. Необходимо рассмотреть все возможные маршруты</p>
9	<p>Один из способов решения: двигаясь слева направо по изображению графа, над каждой вершиной надписывать количество ведущих в неё путей, удовлетворяющих условиям прохождения (непрохождения) через заданные промежуточные вершины</p>	<p>Типичная ошибка – игнорирование в условии задания указаний, что путь должен включать (или не включать) в себя заданную промежуточную точку</p>

Таблица 5

### 3. Алгоритмы и программирование

Номер задания	Рекомендации по выполнению	Типичные ошибки, рекомендации по их предотвращению
5	<p>Рассмотрим типичное задание.</p> <p>У исполнителя Альфа две команды, которым присвоены номера:</p> <p><b>1. прибавь 1</b></p> <p><b>2. умножь на b</b></p> <p>...</p> <p>Известно, что программа <b>11211</b> переводит число 6 в число 82.</p> <p>Определите значение b.</p> <p>Задания такого типа можно решать</p>	<p>Как и в большинстве простых заданий, основные ошибки происходят из-за торопливости и невнимательности.</p> <p>Чтобы уменьшить вероятность ошибки, обязательно после решения делайте проверку.</p> <p>В рассмотренном примере для проверки нужно программу <b>11211</b> с командами</p>

Номер задания	Рекомендации по выполнению	Типичные ошибки, рекомендации по их предотвращению
	<p>составлением линейного уравнения. Запишем программу для заданных чисел в виде уравнения <math>(6 + 1 + 1) \times b + 1 + 1 = 82</math>, отсюда <math>b = 10</math>. Задачу можно решать также перебором. Начнём с конца программы. Заметим, что число 80 (<math>82 - 1 - 1</math>) должно делиться нацело на <math>b</math>. Таким образом, <math>b</math> может принимать только значения 1, 2, 4, 5, 8, 10, 16, 20, 40, 80. Нетрудно заметить, что подходит только 10. Такой способ может оказаться проще, если количество перебираемых множителей невелико, а уравнение не является линейным, т.е. команда умножения встречается в программе более одного раза</p>	<p><b>1. прибавь 1</b>  <b>2. умножь на 10</b>          применить к числу 6 и убедиться, что результатом будет 82</p>
6	<p>При выполнении заданий, подобных примеру из демоверсии, необходимо провести простой анализ программы, выделив условие печати нужного ответа, и определить, при каком значении параметра <math>A</math> указанный результат работы программы будет напечатан необходимое количество раз. Следует отметить, что таких значений может быть несколько. Надо ответить на вопрос, заданный в условии задачи</p>	<p>При выполнении заданий этой линии важно не перепутать логические операции в условии (дизъюнкцию с конъюнкцией), правильно определить, какая ветвь условного оператора выполняется при истинном условии, а также быть внимательным и правильно выполнить строгое/нестрогое сравнение (если оно есть)</p>
15	<p>Сформулируйте последовательно условия движения Робота вдоль объектов и закрашивания им клеток, затем составьте алгоритм, запишите его в виде программы для исполнителя и проверьте правильность её работы в среде разработки «Кумир» (или аналогичной) на различных исходных конфигурациях, соответствующих условию задачи</p>	<p>Обратите внимание на то, что в условии указано: «Алгоритм должен решать задачу для любого допустимого расположения стен и любого расположения и размера проходов внутри стен». Это означает, что нужно обязательно проверить работу алгоритма на различных допустимых конфигурациях, т.е. при различных размерах стен и проходов между ними. То, что программа успешно прошла тестирование, не гарантирует отсутствия в ней ошибок, но существенно снижает их вероятность. Выбирайте в том числе сложные тесты. Помните, что цель тестирования не продемонстрировать самому себе правильность работы программы, а выявить и своевременно устранить возможную ошибку</p>

Номер задания	Рекомендации по выполнению	Типичные ошибки, рекомендации по их предотвращению
16	Внимательно прочитайте описание исходных данных и назначения программы. Сформулируйте алгоритм (поскольку программа простая, это можно сделать в уме, просто проговорив про себя шаги), запишите его в виде программы, протестируйте её на различных исходных значениях, при необходимости внесите исправления. Выбирайте те язык и среду программирования, которыми вы лучше всего владеете. Никакого повышения или снижения баллов за экзотичность языка не предусмотрено	Всё, что написано выше о тестировании программ в рекомендациях к заданию 15, справедливо и для этого задания. Обратите внимание на то, что задания предполагают два вида исходных последовательностей: начинающихся числом, соответствующим количеству элементов последовательности, или неизвестной длины, но заканчивающихся нулём. Последовательности первого типа обрабатываются циклом «for» («для»); последовательности второго типа – циклом «while» («пока»)

Таблица 6

#### 4. Информационные технологии

Номер задания	Рекомендации по выполнению	Типичные ошибки
13	Вы должны выбрать только одно из двух заданий: 13.1 или 13.2. Выбирайте то задание, к которому вы лучше подготовлены. Каждое из них оценивается от 0 до 2 баллов. Обратите внимание на то, что каждый вариант задания 13 устанавливает единственный возможный формат файла ответа. Эти форматы универсальные и поддерживаются всеми современными программами создания презентаций и текстовыми процессорами	
13.1	У этого задания довольно длинное условие. Внимательно прочитайте его, убедитесь, что вам полностью понятны требования к содержанию и оформлению презентации. Внимательно ознакомьтесь с предложенными вам исходными материалами, подумайте, что из них необходимо включить в презентацию. Презентация должна быть краткой и наглядной. Не пытайтесь перенести в неё целиком весь текст, представленный в исходных материалах. Отберите только то, что наиболее соответствует заданной теме. Вы можете отредактировать или переформулировать текст источника, не искажая его смысла. Не следует перегружать слайды разного рода	Типичной ошибкой является попытка включить весь или почти весь текст из исходных материалов в презентацию, поскольку это неизбежно приводит к нарушению заданных в условии требований к оформлению. Не допускайте деформации (искажения пропорций) при размещении изображений. Тщательно проверяйте размеры всех литер в текстовых блоках. Следите, чтобы все тексты в презентации, включая заголовки, были выполнены шрифтом одного типа

	спеце́ффектами. Оформление презентации не должно отвлекать потенциального зрителя от её содержания	
13.2	При выполнении этого задания важно строго соблюсти параметры оформления документа, представленные в образце. Обратите внимание на требование к размеру интервалов между абзацами текста, а также между текстом и таблицей. Начиная с 2026 г использование для задания этих интервалов пустых абзацев считается ошибкой и влечёт снижение балла	Распространённые ошибки оформления, возникающие при выполнении задания: <ul style="list-style-type: none"> <li>• используется шрифт неверного размера;</li> <li>• нет абзацного отступа в первой строке абзаца;</li> <li>• используются символы разрыва строки или конца абзаца для разбиения текста на строки;</li> <li>• абзацный отступ сделан при помощи пробелов</li> </ul>
14	Для выполнения этого задания необходимо уметь записывать приведённые в задании логические условия отбора нужных строк таблицы в виде формул, принятых в электронных таблицах, корректно используя абсолютную и относительную адресацию ячеек. Созданная диаграмма обязательно должна содержать так называемую легенду, т.е. краткое пояснение, какой элемент диаграммы соответствует каким данным. Обычно легенда строится автоматизированно с использованием стандартных средств электронных таблиц. Файлы с исходными данными представлены в формате .ods. Этот формат поддерживается всеми современными программами электронных таблиц	За «ручную», без использования формул, обработку таблицы оценка при верном результате не снижается, но следует учесть, что вероятность допустить ошибку при «ручной» обработке большой таблицы выше

Рекомендуется следующая последовательность действий при подготовке к экзамену.

1. Провести самодиагностику: выполнить задания демонстрационного варианта КИМ и самостоятельно проверить ответы, воспользовавшись эталонными ответами и критериями оценивания. Если есть возможность работать в паре или группой, желательно всегда организовывать взаимную проверку развёрнутых ответов. Цели – выявить собственные пробелы в знаниях и темы, вызвавшие затруднения, зафиксировать исходный уровень подготовки.
2. Заполнить индивидуальный план подготовки к экзамену и следовать ему.
3. При повторении каждой темы сначала выполнять задания по линиям, не менее чем по три-четыре задания каждого типа, встречающегося в линии, затем выполнять задания группами, относящимися к данной теме. После того как ошибки в выполнении заданий по данной теме сведены к минимуму, можно переходить к проработке следующей темы.
4. После повторения всех тем следует выполнить с контролем времени ещё как минимум один вариант КИМ и сравнить результаты (см. п. 1). Также снова следует выявить темы и линии заданий, вызвавшие затруднения, и дополнительно их проработать.

**Разбор заданий ОГЭ прошлых лет  
и упражнения для самостоятельной подготовки**

Рассмотрим примеры выполнения заданий с кратким ответом из линий, вызвавших затруднения у участников ОГЭ.

**Задание 4 (пример)**

Между населёнными пунктами  $A, B, C, D, E$  построены дороги, протяжённость которых (в километрах) приведена в таблице.

	A	B	C	D	E
A		1	4	3	7
B	1		2	5	
C	4	2		3	
D	3	5	3		2
E	7			2	

Определите длину кратчайшего пути между пунктами  $A$  и  $E$ , проходящего через пункт  $C$ . Передвигаться можно только по дорогам, протяжённость которых указана в таблице. Каждый пункт можно посетить не более одного раза.

Решение:

Выпишем все пути, удовлетворяющие условию задачи, и подсчитаем длину каждого из них.

Путь	Длина
ABCDE	$1 + 2 + 3 + 2 = 8$
ACBDE	$4 + 2 + 5 + 2 = 13$
ACDE	$4 + 3 + 2 = 9$

Ответ: 8

**Тренировочные задания линии 4**

Вариант 1

Между населёнными пунктами  $A, B, C, D, E$  построены дороги, протяжённость которых (в километрах) приведена в таблице.

	A	B	C	D	E
A		1	4	3	6
B	1		2	5	
C	4	2		1	
D	3	5	1		2
E	6			2	

Определите длину кратчайшего пути между пунктами  $A$  и  $E$ , проходящего через пункт  $B$ . Передвигаться можно только по дорогам, протяжённость которых указана в таблице. Каждый пункт можно посетить не более одного раза.

Ответ: 6.

### Вариант 2

Между населёнными пунктами  $A, B, C, D, E$  построены дороги, протяжённость которых (в километрах) приведена в таблице.

	A	B	C	D	E
A		6	4	3	6
B	6		2	5	
C	4	2		1	
D	3	5	1		2
E	6			2	

Определите длину кратчайшего пути между пунктами  $A$  и  $E$ , проходящего через пункт  $D$ . Передвигаться можно только по дорогам, протяжённость которых указана в таблице. Каждый пункт можно посетить не более одного раза.

Ответ: 5.

### Вариант 3

Между населёнными пунктами  $A, B, C, D, E$  построены дороги, протяжённость которых (в километрах) приведена в таблице.

	A	B	C	D	E
A		6	4	7	6
B	6		2	5	
C	4	2		1	
D	7	5	1		2
E	6			2	

Определите длину кратчайшего пути между пунктами  $A$  и  $E$ , проходящего через пункт  $D$ . Передвигаться можно только по дорогам, протяжённость которых указана в таблице. Каждый пункт можно посетить не более одного раза.

Ответ: 7.

Вариант 4

Между населёнными пунктами  $A, B, C, D, E$  построены дороги, протяжённость которых (в километрах) приведена в таблице.

	A	B	C	D	E
A		6	4	7	10
B	6		2	5	
C	4	2		1	
D	7	5	1		2
E	10			2	

Определите длину кратчайшего пути между пунктами  $A$  и  $E$ , **не** проходящего через пункт  $B$ . Передвигаться можно только по дорогам, протяжённость которых указана в таблице. Каждый пункт можно посетить не более одного раза.

Ответ: 7.

Вариант 5

Между населёнными пунктами  $A, B, C, D, E$  построены дороги, протяжённость которых (в километрах) приведена в таблице.

	A	B	C	D	E
A		6	4	7	10
B	6		2	5	
C	4	2		1	
D	7	5	1		2
E	10			2	

Определите длину кратчайшего пути между пунктами  $A$  и  $E$ , **не** проходящего через пункт  $C$ . Передвигаться можно только по дорогам, протяжённость которых указана в таблице. Каждый пункт можно посетить не более одного раза.

Ответ: 9.

## Задание 6 (пример)

Ниже приведена программа, записанная на пяти языках программирования.

Алгоритмический язык	Паскаль
<pre>алг нач цел s, t ввод s ввод t если s &gt; 10 или t &gt; 10   то вывод 'YES'   иначе вывод 'NO' все кон</pre>	<pre>var s, t: integer; begin   readln(s);   readln(t);   if (s &gt; 10) or (t &gt; 10)     then       writeln('YES')     else       writeln('NO') end.</pre>
Бейсик	Python
<pre>DIM s, t AS INTEGER INPUT s INPUT t IF s &gt; 10 OR t &gt; 10 THEN   PRINT 'YES' ELSE   PRINT 'NO' ENDIF</pre>	<pre>s = int(input()) t = int(input()) if (s &gt; 10) or (t &gt; 10):   print('YES') else:   print('NO')</pre>
C++	
<pre>#include &lt;iostream&gt; using namespace std;  int main(){   int s, t;   cin &gt;&gt; s;   cin &gt;&gt; t;   if (s &gt; 10    t &gt; 10)     cout &lt;&lt; 'YES' &lt;&lt; endl;   else     cout &lt;&lt; 'NO' &lt;&lt; endl;   return 0; }</pre>	

Было проведено 9 запусков программы, при которых в качестве значений переменных вводились следующие пары чисел  $(s, t)$ :

$(1, 2)$ ;  $(11, 2)$ ;  $(1, 12)$ ;  $(11, 12)$ ;  $(-11, -12)$ ;  $(-11, 12)$ ;  $(-12, 11)$ ;  $(10, 10)$ ;  $(10, 5)$ .

Сколько было запусков, при которых программа напечатала «YES»?

### Решение.

В соответствии с условием оператора ветвления в тексте программы она напечатает «YES» в случае истинности высказывания  $(s > 10)$  **ИЛИ**  $(t > 10)$ .

Подсчитаем количество пар  $(s, t)$ , для которых это высказывание истинно, т.е. хотя бы один из элементов пары больше 10. Ложность высказывания будем обозначать нулём, истинность – единицей.

$(s, t)$	$(s > 10)$ ИЛИ $(t > 10)$
(1, 2)	0
(11, 2)	1
(1, 12)	1
(11, 12)	1
(-11, -12)	0
(-11, 12)	1
(-12, 11)	1
(10, 10)	0
(10, 5)	0

В правом столбце 5 единиц, поэтому ответ – 5.

Ответ: 5.

## Тренировочные задания линии 6

### Вариант 1

Ниже приведена программа, записанная на пяти языках программирования.

Алгоритмический язык	Паскаль
<pre> алг нач цел s, t ввод s ввод t если s &gt;= 10 или t &gt;= 10 то вывод 'YES' иначе вывод 'NO' все кон </pre>	<pre> var s, t: integer; begin   readln(s);   readln(t);   if (s &gt;= 10) or (t &gt;= 10)   then     writeln('YES')   else     writeln('NO') end. </pre>
Бейсик	Python
<pre> DIM s, t AS INTEGER INPUT s INPUT t IF s &gt;= 10 OR t &gt;= 10 THEN   PRINT 'YES' ELSE   PRINT 'NO' ENDIF </pre>	<pre> s = int(input()) t = int(input()) if (s &gt;= 10) or (t &gt;= 10):   print('YES') else:   print('NO') </pre>
C++	
<pre> #include &lt;iostream&gt; using namespace std;  int main(){   int s, t;   cin &gt;&gt; s;   cin &gt;&gt; t;   if (s &gt;= 10    t &gt;= 10)     cout &lt;&lt; 'YES' &lt;&lt; endl;   else     cout &lt;&lt; 'NO' &lt;&lt; endl;   return 0; } </pre>	

Было проведено 9 запусков программы, при которых в качестве значений переменных вводились следующие пары чисел  $(s, t)$ :

(1, 2); (11, 2); (1, 12); (11, 12); (-11, -12); (-11, 12); (-12, 11); (10, 10); (10, 5).

Сколько было запусков, при которых программа напечатала «YES»?

Ответ: 7.

### Вариант 2

Ниже приведена программа, записанная на пяти языках программирования.

<b>Алгоритмический язык</b>	<b>Паскаль</b>
<pre> алг нач цел s, t ввод s ввод t если s &gt; 10 или t &gt;= 10     то вывод 'YES'     иначе вывод 'NO' все кон                     </pre>	<pre> var s, t: integer; begin   readln(s);   readln(t);   if (s &gt; 10) or (t &gt;= 10)     then       writeln('YES')     else       writeln('NO') end.                     </pre>
<b>Бейсик</b>	<b>Python</b>
<pre> DIM s, t AS INTEGER INPUT s INPUT t IF s &gt; 10 OR t &gt;= 10 THEN   PRINT 'YES' ELSE   PRINT 'NO' ENDIF                     </pre>	<pre> s = int(input()) t = int(input()) if (s &gt; 10) or (t &gt;= 10):   print('YES') else:   print('NO')                     </pre>
<b>C++</b>	
<pre> #include &lt;iostream&gt; using namespace std;  int main(){   int s, t;   cin &gt;&gt; s;   cin &gt;&gt; t;   if (s &gt; 10    t &gt;= 10)     cout &lt;&lt; 'YES' &lt;&lt; endl;   else     cout &lt;&lt; 'NO' &lt;&lt; endl;   return 0; }                     </pre>	

Было проведено 9 запусков программы, при которых в качестве значений переменных вводились следующие пары чисел  $(s, t)$ :

(1, 2); (11, 2); (1, 12); (11, 12); (-11, -12); (-11, 12); (-12, 11); (10, 10); (10, 5).

Сколько было запусков, при которых программа напечатала «YES»?

Ответ: 6.

### Вариант 3

Ниже приведена программа, записанная на пяти языках программирования.

Алгоритмический язык	Паскаль
<pre>алг нач цел s, t ввод s ввод t если s &gt; 10 и t &gt; 10     то вывод 'YES'     иначе вывод 'NO' все кон</pre>	<pre>var s, t: integer; begin   readln(s);   readln(t);   if (s &gt; 10) and (t &gt; 10)     then       writeln('YES')     else       writeln('NO') end.</pre>
Бейсик	Python
<pre>DIM s, t AS INTEGER INPUT s INPUT t IF s &gt; 10 AND t &gt; 10 THEN   PRINT 'YES' ELSE   PRINT 'NO' ENDIF</pre>	<pre>s = int(input()) t = int(input()) if (s &gt; 10) and (t &gt; 10):   print('YES') else:   print('NO')</pre>
C++	
<pre>#include &lt;iostream&gt; using namespace std;  int main(){   int s, t;   cin &gt;&gt; s;   cin &gt;&gt; t;   if (s &gt; 10 &amp;&amp; t &gt; 10)     cout &lt;&lt; 'YES' &lt;&lt; endl;   else     cout &lt;&lt; 'NO' &lt;&lt; endl;   return 0; }</pre>	

Было проведено 9 запусков программы, при которых в качестве значений переменных вводились следующие пары чисел  $(s, t)$ :

(1, 2); (11, 2); (1, 12); (11, 12); (-11, -12); (-11, 12); (-12, 11); (10, 10); (10, 5).

Сколько было запусков, при которых программа напечатала «YES»?

Ответ: 1.

#### Вариант 4

Ниже приведена программа, записанная на пяти языках программирования.

Алгоритмический язык	Паскаль
<pre>алг нач цел s, t ввод s ввод t если s &gt;= 10 и t &gt;= 10     то вывод 'YES'     иначе вывод 'NO' все кон</pre>	<pre>var s, t: integer; begin   readln(s);   readln(t);   if (s &gt;= 10) and (t &gt;= 10)     then       writeln('YES')     else       writeln('NO') end.</pre>
Бейсик	Python
<pre>DIM s, t AS INTEGER INPUT s INPUT t IF s &gt;= 10 AND t &gt;= 10 THEN   PRINT 'YES' ELSE   PRINT 'NO' ENDIF</pre>	<pre>s = int(input()) t = int(input()) if (s &gt;= 10) and (t &gt;= 10):   print('YES') else:   print('NO')</pre>
C++	
<pre>#include &lt;iostream&gt; using namespace std;  int main(){   int s, t;   cin &gt;&gt; s;   cin &gt;&gt; t;   if (s &gt;= 10 &amp;&amp; t &gt;= 10)     cout &lt;&lt; 'YES' &lt;&lt; endl;   else     cout &lt;&lt; 'NO' &lt;&lt; endl;   return 0; }</pre>	

Было проведено 9 запусков программы, при которых в качестве значений переменных вводились следующие пары чисел  $(s, t)$ :

(1, 2); (11, 2); (1, 12); (11, 12); (-11, -12); (-11, 12); (-12, 11); (10, 10); (10, 5).

Сколько было запусков, при которых программа напечатала «NO»?

Ответ: 7.

## Вариант 5

Ниже приведена программа, записанная на пяти языках программирования.

Алгоритмический язык	Паскаль
<pre>алг нач цел s, t ввод s ввод t если s &lt;= 10 и t &lt;= 10     то вывод 'YES'     иначе вывод 'NO' все кон</pre>	<pre>var s, t: integer; begin   readln(s);   readln(t);   if (s &lt;= 10) and (t &lt;= 10)     then       writeln('YES')     else       writeln('NO') end.</pre>
Бейсик	Python
<pre>DIM s, t AS INTEGER INPUT s INPUT t IF s &lt;= 10 AND t &lt;= 10 THEN   PRINT 'YES' ELSE   PRINT 'NO' ENDIF</pre>	<pre>s = int(input()) t = int(input()) if (s &lt;= 10) and (t &lt;= 10):   print('YES') else:   print('NO')</pre>
C++	
<pre>#include &lt;iostream&gt; using namespace std;  int main(){   int s, t;   cin &gt;&gt; s;   cin &gt;&gt; t;   if (s &lt;= 10 &amp;&amp; t &lt;= 10)     cout &lt;&lt; 'YES' &lt;&lt; endl;   else     cout &lt;&lt; 'NO' &lt;&lt; endl;   return 0; }</pre>	

Было проведено 9 запусков программы, при которых в качестве значений переменных вводились следующие пары чисел  $(s, t)$ :

(1, 2); (11, 2); (1, 12); (11, 12); (-11, -12); (-11, 12); (-12, 11); (10, 10); (10, 5).

Сколько было запусков, при которых программа напечатала «NO»?

Ответ: 5.

### Задание 8 (демонстрационный вариант 2024 г.)

В языке запросов поискового сервера для обозначения логической операции «ИЛИ» используется символ «|», а для обозначения логической операции «И» – символ «&».

В таблице приведены запросы и количество найденных по ним страниц некоторого сегмента сети Интернет.

Запрос	Найдено страниц (в тысячах)
<i>Волга &amp; (Ока   Кама)</i>	505
<i>Волга &amp; Ока</i>	230
<i>Волга &amp; Кама</i>	400

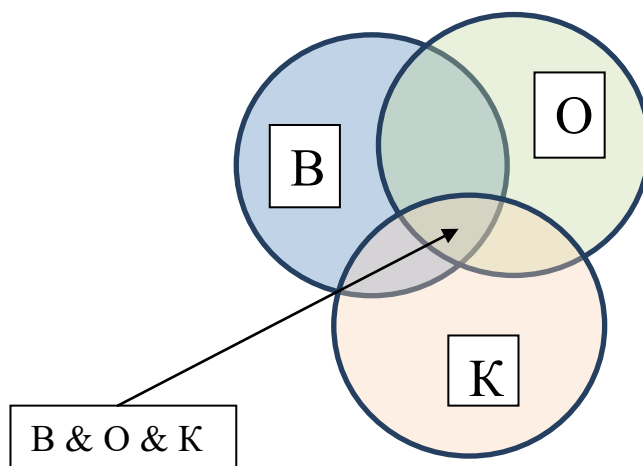
Какое количество страниц (в тысячах) будет найдено по запросу  
*Волга & Ока & Кама*?

Считается, что все запросы выполнялись практически одновременно, так что набор страниц, содержащих все искомые слова, не изменялся за время выполнения запросов.

Решение.

Обозначим результат запроса *Волга* буквой *В*, *Ока* – *О*, *Кама* – *К*.

Построим диаграмму Венна.



Обозначим  $\text{Card}(x)$  мощность множества  $x$ .

Тогда, исходя из построенной диаграммы,

$$\text{Card}(B \& O \& K) = \text{Card}(B \& O) + \text{Card}(B \& K) - \text{Card}(B \& (O / K)) = 125.$$

Ответ: 125.

## Тренировочные задания линии 8

### Вариант 1

В языке запросов поискового сервера для обозначения логической операции «ИЛИ» используется символ «|», а для обозначения логической операции «И» – символ «&».

В таблице приведены запросы и количество найденных по ним страниц некоторого сегмента сети Интернет.

Запрос	Найдено страниц (в тысячах)
<i>Волга &amp; (Ока   Кама)</i>	505
<i>Волга &amp; Кама</i>	400
<i>Волга &amp; Ока &amp; Кама</i>	120

Какое количество страниц (в тысячах) будет найдено по запросу  
*Волга & Ока*?

Считается, что все запросы выполнялись практически одновременно, так что набор страниц, содержащих все искомые слова, не изменялся за время выполнения запросов.

Ответ: 225.

### Вариант 2

В языке запросов поискового сервера для обозначения логической операции «ИЛИ» используется символ «|», а для обозначения логической операции «И» – символ «&».

В таблице приведены запросы и количество найденных по ним страниц некоторого сегмента сети Интернет.

Запрос	Найдено страниц (в тысячах)
<i>Волга &amp; Ока</i>	240
<i>Волга &amp; Кама</i>	400
<i>Волга &amp; Ока &amp; Кама</i>	120

Какое количество страниц (в тысячах) будет найдено по запросу  
*Волга & (Ока | Кама)*?

Считается, что все запросы выполнялись практически одновременно, так что набор страниц, содержащих все искомые слова, не изменялся за время выполнения запросов.

Ответ: 520.

### Вариант 3

В языке запросов поискового сервера для обозначения логической операции «ИЛИ» используется символ «|», а для обозначения логической операции «И» – символ «&».

В таблице приведены запросы и количество найденных по ним страниц некоторого сегмента сети Интернет.

<b>Запрос</b>	<b>Найдено страниц (в тысячах)</b>
<i>Ока &amp; (Волга   Кама)</i>	510
<i>Волга &amp; Ока</i>	230
<i>Ока &amp; Кама</i>	400

Какое количество страниц (в тысячах) будет найдено по запросу  
*Волга & Ока & Кама*?

Считается, что все запросы выполнялись практически одновременно, так что набор страниц, содержащих все искомые слова, не изменялся за время выполнения запросов.

Ответ: 120.

### Вариант 4

В языке запросов поискового сервера для обозначения логической операции «ИЛИ» используется символ «|», а для обозначения логической операции «И» – символ «&».

В таблице приведены запросы и количество найденных по ним страниц некоторого сегмента сети Интернет.

<b>Запрос</b>	<b>Найдено страниц (в тысячах)</b>
<i>Ока &amp; (Волга   Кама)</i>	510
<i>Волга &amp; Ока</i>	230
<i>Волга &amp; Ока &amp; Кама</i>	115

Какое количество страниц (в тысячах) будет найдено по запросу  
*Ока & Кама*?

Считается, что все запросы выполнялись практически одновременно, так что набор страниц, содержащих все искомые слова, не изменялся за время выполнения запросов.

Ответ: 395.

### Вариант 5

В языке запросов поискового сервера для обозначения логической операции «ИЛИ» используется символ «|», а для обозначения логической операции «И» – символ «&».

В таблице приведены запросы и количество найденных по ним страниц некоторого сегмента сети Интернет.

Запрос	Найдено страниц (в тысячах)
<i>Ока &amp; Кама</i>	390
<i>Волга &amp; Ока</i>	230
<i>Волга &amp; Ока &amp; Кама</i>	115

Какое количество страниц (в тысячах) будет найдено по запросу  
*Ока & (Волга | Кама)*?

Считается, что все запросы выполнялись практически одновременно, так что набор страниц, содержащих все искомые слова, не изменялся за время выполнения запросов.

Ответ: 505.

### **Задание 10 (пример)**

Среди приведённых ниже трёх чисел, записанных в различных системах счисления, найдите максимальное и запишите его в ответе в десятичной системе счисления. В ответе запишите только число, основание системы счисления указывать не нужно.

$23_{16}$ ,  $32_8$ ,  $11110_2$

Решение (1-й способ).

Запишем все три числа в десятичной системе счисления.

$$23_{16} = 2 \cdot 16 + 3 = 35$$

$$32_8 = 3 \cdot 8 + 2 = 26$$

$$11110_2 = 16 + 8 + 4 + 2 + 0 = 30$$

Ответ: 35.

Решение (2-й способ).

Сравним все три числа в двоичной системе счисления, используя для перевода из шестнадцатеричной системы счисления в двоичную деление на тетрады, а из восьмеричной деление на триады.

$$23_{16} = 0010\ 0011_2$$

$$32_8 = 011\ 010_2$$

$$11110_2$$

Видно, что максимальное число –  $0010\ 0011_2$ . Оно же  $23_{16} = 2 \cdot 16 + 3 = 35$ .

Ответ: 35.

## Тренировочные задания линии 10

### Вариант 1

Среди приведённых ниже трёх чисел, записанных в различных системах счисления, найдите максимальное и запишите его в ответе в десятичной системе счисления. В ответе запишите только число, основание системы счисления указывать не нужно.

$$F0_{16}, 77_8, 11111110_2$$

Ответ: 254.

### Вариант 2

Среди приведённых ниже трёх чисел, записанных в различных системах счисления, найдите минимальное и запишите его в ответе в десятичной системе счисления. В ответе запишите только число, основание системы счисления указывать не нужно.

$$F0_{16}, 77_8, 11111110_2$$

Ответ: 63.

### Вариант 3

Вычислите значение выражения и запишите его в ответе в десятичной системе счисления. В ответе запишите только число, основание системы счисления указывать не нужно.

$$(A0_{16} - 12_8) / 10_2$$

Ответ: 75.

### Вариант 4

Вычислите значение выражения и запишите его в ответе в десятичной системе счисления. В ответе запишите только число, основание системы счисления указывать не нужно.

$$(FF_{16} - 17_8) / 101_2$$

Ответ: 48.

### Вариант 5

Вычислите значение выражения и запишите его в ответе в десятичной системе счисления. В ответе запишите только число, основание системы счисления указывать не нужно.

$$(2F_{16} + 11_8) \cdot 11_2$$

Ответ: 168.

### Задание 15 (пример)

Исполнитель Робот умеет перемещаться по лабиринту, начерченному на плоскости, разбитой на клетки. Между соседними (по сторонам) клетками может стоять стена, через которую Робот пройти не может.

У Робота есть девять команд. Четыре команды – это команды-приказы:

**вверх вниз влево вправо**

При выполнении любой из этих команд Робот перемещается на одну клетку соответственно: вверх ↑, вниз ↓, влево ←, вправо →. Если Робот получит команду передвижения сквозь стену, то он разрушится.

Также у Робота есть команда **закрасить**, при которой закрашивается клетка, в которой Робот находится в настоящий момент.

Ещё четыре команды – это команды проверки условий. Эти команды проверяют, свободен ли путь для Робота в каждом из четырёх возможных направлений:

**сверху свободно снизу свободно слева свободно справа свободно**

Эти команды можно использовать вместе с условием «если», имеющим следующий вид:

**если условие то**

*последовательность команд*

**все**

Здесь *условие* – одна из команд проверки условия.

*Последовательность команд* – это одна или несколько любых команд-приказов.

Например, для передвижения на одну клетку вправо, если справа нет стенки и закрашивания клетки, можно использовать такой алгоритм:

**если справа свободно то**

**вправо**

**закрасить**

**все**

В одном условии можно использовать несколько команд проверки условий, применяя логические связки **и**, **или**, **не**, например:

**если (справа свободно) и (не снизу свободно) то**

**вправо**

**все**

Для повторения последовательности команд можно использовать цикл «пока», имеющий следующий вид:

**нц пока условие**

*последовательность команд*

**кц**

Например, для движения вправо, пока это возможно, можно использовать следующий алгоритм:

**нц пока справа свободно**

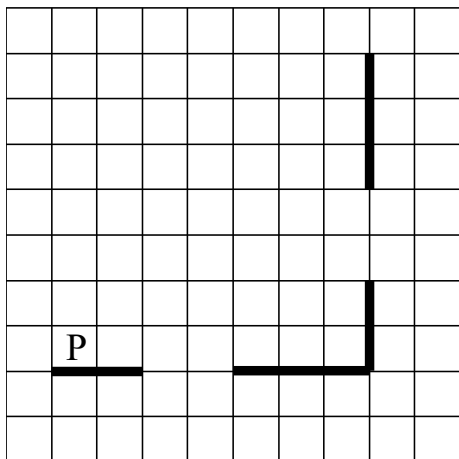
**вправо**

**кц**

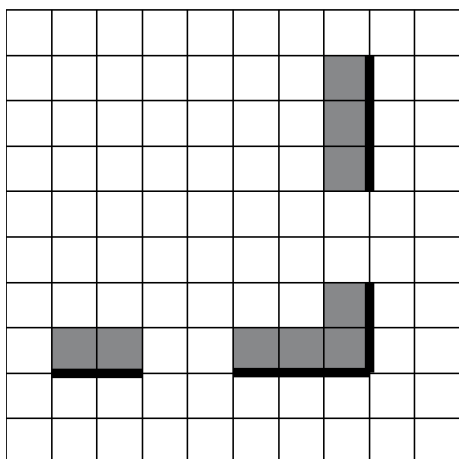
**Выполните задание.**

На бесконечном поле есть горизонтальная и вертикальная стены. Правый конец горизонтальной стены соединён с нижним концом вертикальной стены. **Длины стен неизвестны.** В каждой стене есть ровно один проход, точное место прохода и его ширина неизвестны. Робот находится в клетке, расположенной непосредственно над горизонтальной стеной у её левого конца.

На рисунке указан один из возможных способов расположения стен и Робота (Робот обозначен буквой «Р»).



Напишите для Робота алгоритм, закрашивающий все клетки, расположенные непосредственно выше горизонтальной стены и левее вертикальной стены. Проходы должны остаться незакрашенными. Робот должен закрасить только клетки, удовлетворяющие данному условию. Например, для приведённого выше рисунка Робот должен закрасить следующие клетки (см. рисунок).



При исполнении алгоритма Робот не должен разрушиться, выполнение алгоритма должно завершиться. Конечное положение Робота может быть произвольным.

Алгоритм должен решать задачу для любого допустимого расположения стен и любого расположения и размера проходов внутри стен.

Алгоритм может быть выполнен в среде формального исполнителя или записан в текстовом редакторе.

Сохраните алгоритм в формате программы Кумир или в текстовом файле. Название файла и каталог для сохранения Вам сообщат организаторы экзамена.

### Содержание верного ответа

(допускаются иные формулировки ответа, не искажающие его смысла)

Команды исполнителя будем записывать жирным шрифтом, а комментарии, поясняющие алгоритм и не являющиеся его частью, – курсивом. Начало комментария будем обозначать символом «|».

| *Двигаемся вправо, пока не дойдём до прохода в горизонтальной стене, и закрашиваем клетки.*

**нц пока не снизу свободно**

**закрасить**

**вправо**

**кц**

| *Двигаемся дальше до горизонтальной стены.*

**нц пока снизу свободно**

**вправо**

**кц**

| *Двигаемся вправо, пока не дойдём до вертикальной стены, и закрашиваем клетки.*

**нц пока справа свободно**

**закрасить**

**вправо**

**кц**

| *Двигаемся вверх, пока не дойдём до прохода в вертикальной стене, и закрашиваем клетки.*

**нц пока не справа свободно**

**закрасить**

**вверх**

**кц**

| *Двигаемся дальше до вертикальной стены.*

**нц пока справа свободно**

**вверх**

**кц**

| *Двигаемся вверх до конца вертикальной стены и закрашиваем клетки.*

**нц пока не справа свободно**

**закрасить**

**вверх**

**кц**

Возможны и другие варианты решения.

Допускается использование иного синтаксиса инструкций исполнителя, более привычного для учащихся, в частности использование проверки «справа стена» вместо «не справа свободно».

Допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора решения

Указания по оцениванию	Баллы
Алгоритм правильно работает при всех допустимых исходных данных	2
При всех допустимых исходных данных верно следующее: 1) выполнение алгоритма завершается, и при этом Робот не разбивается; 2) закрашено не более 10 лишних клеток; 3) остались незакрашенными не более 10 клеток из числа тех, которые должны были быть закрашены	1
Задание выполнено неверно, т.е. не выполнены условия, позволяющие поставить 1 или 2 балла	0
<i>Максимальный балл</i>	2

## Тренировочные задания линии 15

### Вариант 1

Исполнитель Робот умеет перемещаться по лабиринту, начерченному на плоскости, разбитой на клетки. Между соседними (по сторонам) клетками может стоять стена, через которую Робот пройти не может.

У Робота есть девять команд. Четыре команды – это команды-приказы:

**вверх   вниз   влево   вправо**

При выполнении любой из этих команд Робот перемещается на одну клетку соответственно: вверх ↑, вниз ↓, влево ←, вправо →. Если Робот получит команду передвижения сквозь стену, то он разрушится.

Также у Робота есть команда **закрасить**, при которой закрашивается клетка, в которой Робот находится в настоящий момент.

Ещё четыре команды – это команды проверки условий. Эти команды проверяют, свободен ли путь для Робота в каждом из четырёх возможных направлений:

**сверху свободно   снизу свободно   слева свободно   справа свободно**

Эти команды можно использовать вместе с условием «если», имеющим следующий вид:

**если условие то**

*последовательность команд*

**все**

Здесь *условие* – одна из команд проверки условия.

*Последовательность команд* – это одна или несколько любых команд-приказов.

Например, для передвижения на одну клетку вправо, если справа нет стенки, и закрашивания клетки можно использовать такой алгоритм:

**если справа свободно то**

**вправо**

**закрасить**

**все**

В одном условии можно использовать несколько команд проверки условий, применяя логические связки **и**, **или**, **не**, например:

**если (справа свободно) и (не снизу свободно) то**

**вправо**

**все**

Для повторения последовательности команд можно использовать цикл «пока», имеющий следующий вид:

**нц пока условие**

*последовательность команд*

**кц**

Например, для движения вправо, пока это возможно, можно использовать следующий алгоритм:

**нц пока справа свободно**

**вправо**

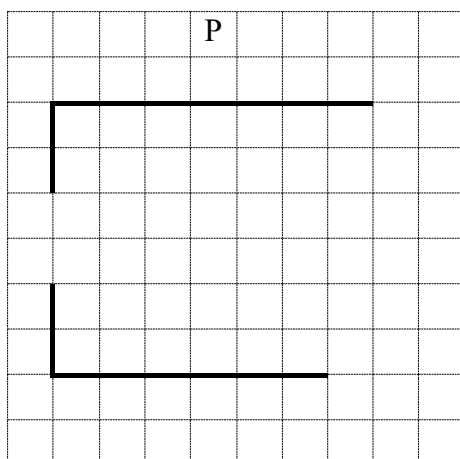
**кц**

**Выполните задание.**

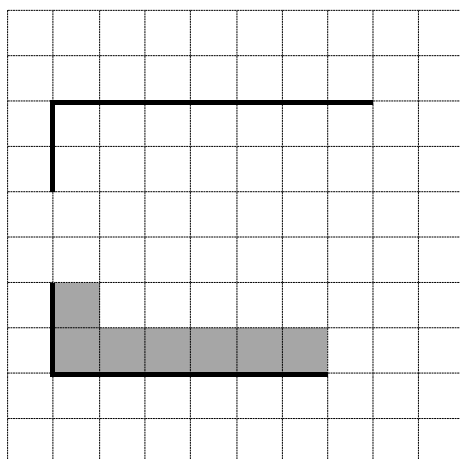
На бесконечном поле есть две горизонтальные стены и одна вертикальная. Левый конец первой горизонтальной стены соединён с верхним концом вертикальной стены, а нижний конец вертикальной стены соединён с левым концом второй горизонтальной стены. **Длины стен неизвестны.** В вертикальной стене есть ровно один проход, точное место прохода

и его ширина неизвестны. Робот находится в одной из клеток, расположенной над верхней стеной.

На рисунке указан один из возможных способов расположения стен и Робота (Робот обозначен буквой «Р»).



Напишите для Робота алгоритм, закрашивающий все клетки, расположенные непосредственно над второй горизонтальной стеной, и клетки, расположенные непосредственно правее вертикальной стены, но ниже прохода в ней. Робот должен закрасить только клетки, удовлетворяющие данному условию. Например, для приведённого выше рисунка Робот должен закрасить следующие клетки (см. рисунок).



При исполнении алгоритма Робот не должен разрушиться, выполнение алгоритма должно завершиться. Конечное положение Робота может быть произвольным.

Алгоритм должен решать задачу для любого допустимого расположения стен и любого расположения и размера прохода внутри стены.

Алгоритм может быть выполнен в среде формального исполнителя или записан в текстовом редакторе.

Сохраните алгоритм в формате программы Кумир или в текстовом файле. Название файла и каталог для сохранения Вам сообщат организаторы экзамена.

**Содержание верного ответа и указания по оцениванию**  
(допускаются иные формулировки ответа, не искажающие его смысла)

Команды исполнителя будем записывать жирным шрифтом, а комментарии, поясняющие алгоритм и не являющиеся его частью, – курсивом. Начало комментария будем обозначать символом «|».

| *Опускаемся вниз до первой горизонтальной стены.*

**нц пока снизу свободно**

**вниз**

**кц**

| *Двигаемся влево до конца первой горизонтальной стены.*

**нц пока не снизу свободно**

**влево**

**кц**

| *Сдвигаемся в клетку у верхнего конца вертикальной стены.*

**вниз**

| *Опускаемся вниз до прохода в вертикальной стене.*

**нц пока не справа свободно**

**вниз**

**кц**

| *Опускаемся вниз до конца прохода в стене.*

**вправо**

**нц пока слева свободно**

**вниз**

**кц**

| *Закрашиваем клетки непосредственно правее вертикальной стены ниже прохода в ней (кроме самой нижней клетки).*

**нц пока снизу свободно**

**закрасить**

**вниз**

**кц**

| *Закрашиваем клетки непосредственно над горизонтальной стеной.*

**нц пока не снизу свободно**

**закрасить**

**вправо**

**кц**

Указания по оцениванию	Баллы
Алгоритм правильно работает при всех допустимых исходных данных	2
При всех допустимых исходных данных верно следующее: 1) выполнение алгоритма завершается, и при этом Робот не разбивается; 2) закрашено не более 10 лишних клеток; 3) остались незакрашенными не более 10 клеток из числа тех, которые должны были быть закрашены	1
Задание выполнено неверно, т.е. не выполнены условия, позволяющие поставить 1 или 2 балла	0
<i>Максимальный балл</i>	2

## Вариант 2

Исполнитель Робот умеет перемещаться по лабиринту, начерченному на плоскости, разбитой на клетки. Между соседними (по сторонам) клетками может стоять стена, через которую Робот пройти не может.

У Робота есть девять команд. Четыре команды – это команды-приказы:

**вверх вниз влево вправо**

При выполнении любой из этих команд Робот перемещается на одну клетку соответственно: вверх ↑, вниз ↓, влево ←, вправо →. Если Робот получит команду передвижения сквозь стену, то он разрушится.

Также у Робота есть команда **закрасить**, при которой закрашивается клетка, в которой Робот находится в настоящий момент.

Ещё четыре команды – это команды проверки условий. Эти команды проверяют, свободен ли путь для Робота в каждом из четырёх возможных направлений:

**сверху свободно снизу свободно слева свободно справа свободно**

Эти команды можно использовать вместе с условием «если», имеющим следующий вид:

**если условие то**

*последовательность команд*

**все**

Здесь *условие* – одна из команд проверки условия.

*Последовательность команд* – это одна или несколько любых команд-приказов.

Например, для передвижения на одну клетку вправо, если справа нет стенки, и закрашивания клетки можно использовать такой алгоритм:

**если справа свободно то**

**вправо**

**закрасить**

**все**

В одном условии можно использовать несколько команд проверки условий, применяя логические связки **и**, **или**, **не**, например:

**если (справа свободно) и (не снизу свободно) то**

**вправо**

**все**

Для повторения последовательности команд можно использовать цикл «**пока**», имеющий следующий вид:

**нц пока условие**

*последовательность команд*

**кц**

*Например*, для движения вправо, пока это возможно, можно использовать следующий алгоритм:

**нц пока справа свободно**

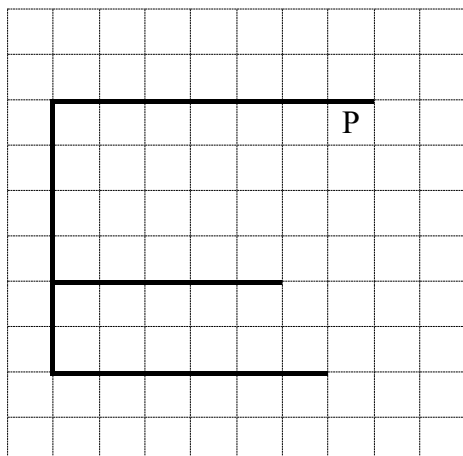
**вправо**

**кц**

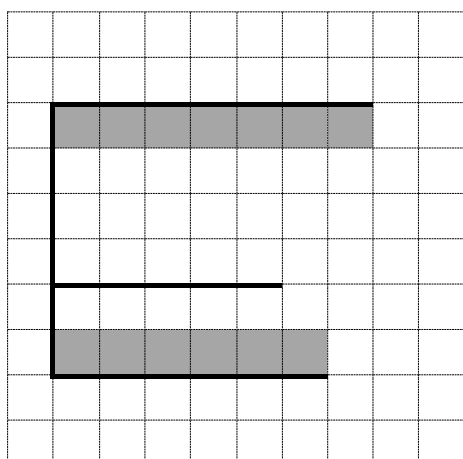
**Выполните задание.**

На бесконечном поле есть три горизонтальные стены и одна вертикальная. Левые концы первой и второй горизонтальных стен соединены с верхним и нижним концами вертикальной стены соответственно, а левый конец третьей горизонтальной стены упирается в вертикальную стену. **Длины стен неизвестны.** Робот находится в клетке, расположенной непосредственно под первой горизонтальной стеной у её правого конца.

На рисунке указан один из возможных способов расположения стен и Робота (Робот обозначен буквой «Р»).



Напишите для Робота алгоритм, закрашивающий все клетки, расположенные непосредственно под первой горизонтальной стеной, и клетки, расположенные непосредственно над второй горизонтальной стеной. Робот должен закрасить только клетки, удовлетворяющие данному условию. Например, для приведённого выше рисунка Робот должен закрасить следующие клетки (см. рисунок).



При исполнении алгоритма Робот не должен разрушиться, выполнение алгоритма должно завершиться. Конечное расположение Робота может быть произвольным.

Алгоритм должен решать задачу для любого допустимого расположения стен.

Алгоритм может быть выполнен в среде формального исполнителя или записан в текстовом редакторе.

Сохраните алгоритм в формате программы Кумир или в текстовом файле. Название файла и каталог для сохранения Вам сообщат организаторы экзамена.

**Содержание верного ответа и указания по оцениванию**  
(допускаются иные формулировки ответа, не искажающие его смысла)

Команды исполнителя будем записывать жирным шрифтом, а комментарии, поясняющие алгоритм и не являющиеся его частью, – курсивом. Начало комментария будем обозначать символом «|».

| *Закрашиваем клетки, сдвигаясь влево к вертикальной стене.*

**нц пока слева свободно**

**закрасить**

**влево**

**кц**

**закрасить**

| *Опускаемся вниз до третьей горизонтальной стены.*

**нц пока снизу свободно**

**вниз**

**кц**

| *Двигаемся вправо до конца третьей горизонтальной стены.*

**нц пока не снизу свободно**

**вправо**

**кц**

| *Сдвигаемся на одну клетку вниз.*

**вниз**

| *Двигаемся влево до конца третьей горизонтальной стены.*

**нц пока слева свободно**

**влево**

**кц**

| *Опускаемся вниз до второй горизонтальной стены.*

**нц пока снизу свободно**

**вниз**

**кц**

| *Закрашиваем клетки, передвигаясь вправо к концу второй горизонтальной стены.*

**нц пока не снизу свободно**

**закрасить**

**вправо**

**кц**

**Указания по оцениванию**

**Баллы**

Алгоритм правильно работает при всех допустимых исходных данных

2

При всех допустимых исходных данных верно следующее:

1

1) выполнение алгоритма завершается, и при этом Робот не разбивается;

2) закрашено не более трёх лишних клеток;

3) остались незакрашенными не более трёх клеток из числа тех, которые должны были быть закрашены

Задание выполнено неверно, т.е. не выполнены условия, позволяющие поставить 1 или 2 балла

0

*Максимальный балл*

2

### Вариант 3

Исполнитель Робот умеет перемещаться по лабиринту, начерченному на плоскости, разбитой на клетки. Между соседними (по сторонам) клетками может стоять стена, через которую Робот пройти не может.

У Робота есть девять команд. Четыре команды – это команды-приказы:

**вверх вниз влево вправо**

При выполнении любой из этих команд Робот перемещается на одну клетку соответственно: вверх ↑, вниз ↓, влево ←, вправо →. Если Робот получит команду передвижения сквозь стену, то он разрушится.

Также у Робота есть команда **закрасить**, при которой закрашивается клетка, в которой Робот находится в настоящий момент.

Ещё четыре команды – это команды проверки условий. Эти команды проверяют, свободен ли путь для Робота в каждом из четырёх возможных направлений:

**сверху свободно снизу свободно слева свободно справа свободно**

Эти команды можно использовать вместе с условием «если», имеющим следующий вид:

**если условие то**

*последовательность команд*

**все**

Здесь *условие* – одна из команд проверки условия.

*Последовательность команд* – это одна или несколько любых команд-приказов.

Например, для передвижения на одну клетку вправо, если справа нет стенки, и закрашивания клетки можно использовать такой алгоритм:

**если справа свободно то**

**вправо**

**закрасить**

**все**

В одном условии можно использовать несколько команд проверки условий, применяя логические связки **и**, **или**, **не**, например:

**если (справа свободно) и (не снизу свободно) то**

**вправо**

**все**

Для повторения последовательности команд можно использовать цикл «пока», имеющий следующий вид:

**нц пока условие**

*последовательность команд*

**кц**

*Например*, для движения вправо, пока это возможно, можно использовать следующий алгоритм:

**нц пока справа свободно**

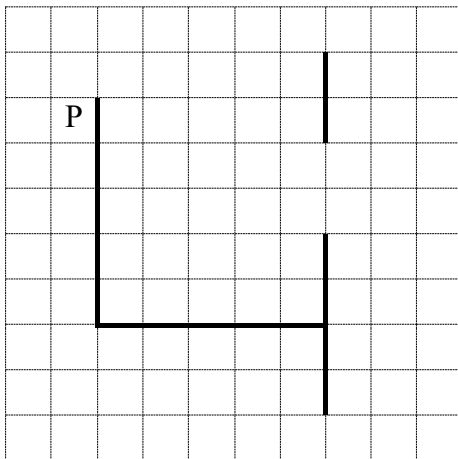
**вправо**

**кц**

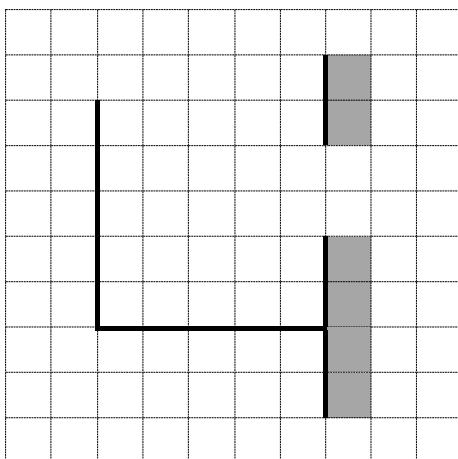
**Выполните задание.**

На бесконечном поле есть две вертикальные стены и одна горизонтальная. Левый конец горизонтальной стены соединён с нижним концом первой вертикальной стены, а правый её конец упирается во вторую вертикальную стену. **Длины стен неизвестны.** Во второй вертикальной стене есть ровно один проход, расположенный строго выше клеток, непосредственно примыкающих сверху к горизонтальной стене, однако точное место прохода и его ширина неизвестны. Робот находится в клетке, расположенной непосредственно левее первой вертикальной стены у её верхнего конца.

На рисунке указан один из возможных способов расположения стен и Робота (Робот обозначен буквой «Р»).



Напишите для Робота алгоритм, закрашивающий все клетки, расположенные непосредственно правее второй вертикальной стены. Проход должен остаться незакрашенным. Робот должен закрасить только клетки, удовлетворяющие данному условию. Например, для приведённого выше рисунка Робот должен закрасить следующие клетки (см. рисунок).



При исполнении алгоритма Робот не должен разрушиться, выполнение алгоритма должно завершиться. Конечное расположение Робота может быть произвольным.

Алгоритм должен решать задачу для любого допустимого расположения стен и любого расположения и размера прохода внутри стены.

Алгоритм может быть выполнен в среде формального исполнителя или записан в текстовом редакторе.

Сохраните алгоритм в формате программы Кумир или в текстовом файле. Название файла и каталог для сохранения Вам сообщат организаторы экзамена.

**Содержание верного ответа и указания по оцениванию**  
(допускаются иные формулировки ответа, не искажающие его смысла)

Команды исполнителя будем записывать жирным шрифтом, а комментарии, поясняющие алгоритм и не являющиеся его частью, – курсивом. Начало комментария будем обозначать символом «|».

| *Опускаемся вниз до конца первой вертикальной стены.*

**нц пока не справа свободно**

**вниз**

**кц**

| *Двигаемся вправо до второй вертикальной стены.*

**нц пока справа свободно**

**вправо**

**кц**

| *Опускаемся вниз до конца второй вертикальной стены.*

**нц пока не справа свободно**

**вниз**

**кц**

| *Сдвигаемся в клетку правее нижнего конца второй вертикальной стены.*

**вправо**

**вверх**

| *Закрашиваем клетки и двигаемся вверх до прохода в стене.*

**нц пока не слева свободно**

**закрасить**

**вверх**

**кц**

| *Двигаемся вверх до конца прохода в стене.*

**нц пока слева свободно**

**вверх**

**кц**

| *Закрашиваем клетки и двигаемся вверх до конца вертикальной стены.*

**нц пока не слева свободно**

**закрасить**

**вверх**

**кц**

<b>Указания по оцениванию</b>	<b>Баллы</b>
Алгоритм правильно работает при всех допустимых исходных данных	2
При всех допустимых исходных данных верно следующее: 1) выполнение алгоритма завершается, и при этом Робот не разбивается; 2) закрашено не более трёх лишних клеток; 3) остались незакрашенными не более трёх клеток из числа тех, которые должны были быть закрашены	1
Задание выполнено неверно, т.е. не выполнены условия, позволяющие поставить 1 или 2 балла	0
<i>Максимальный балл</i>	2

## Вариант 4

Исполнитель Робот умеет перемещаться по лабиринту, начерченному на плоскости, разбитой на клетки. Между соседними (по сторонам) клетками может стоять стена, через которую Робот пройти не может.

У Робота есть девять команд. Четыре команды – это команды-приказы:

**вверх вниз влево вправо**

При выполнении любой из этих команд Робот перемещается на одну клетку соответственно: вверх ↑, вниз ↓, влево ←, вправо →. Если Робот получит команду передвижения сквозь стену, то он разрушится.

Также у Робота есть команда **закрасить**, при которой закрашивается клетка, в которой Робот находится в настоящий момент.

Ещё четыре команды – это команды проверки условий. Эти команды проверяют, свободен ли путь для Робота в каждом из четырёх возможных направлений:

**сверху свободно снизу свободно слева свободно справа свободно**

Эти команды можно использовать вместе с условием «если», имеющим следующий вид:

**если условие то**

*последовательность команд*

**все**

Здесь *условие* – одна из команд проверки условия.

*Последовательность команд* – это одна или несколько любых команд-приказов.

Например, для передвижения на одну клетку вправо, если справа нет стенки, и закрашивания клетки можно использовать такой алгоритм:

**если справа свободно то**

**вправо**

**закрасить**

**все**

В одном условии можно использовать несколько команд проверки условий, применяя логические связки **и**, **или**, **не**, например:

**если (справа свободно) и (не снизу свободно) то**

**вправо**

**все**

Для повторения последовательности команд можно использовать цикл «пока», имеющий следующий вид:

**нц пока условие**

*последовательность команд*

**кц**

*Например*, для движения вправо, пока это возможно, можно использовать следующий алгоритм:

**нц пока справа свободно**

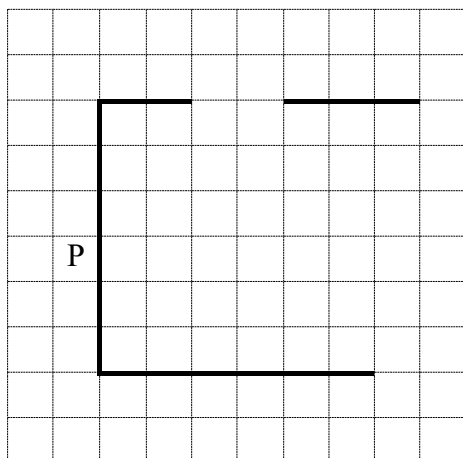
**вправо**

**кц**

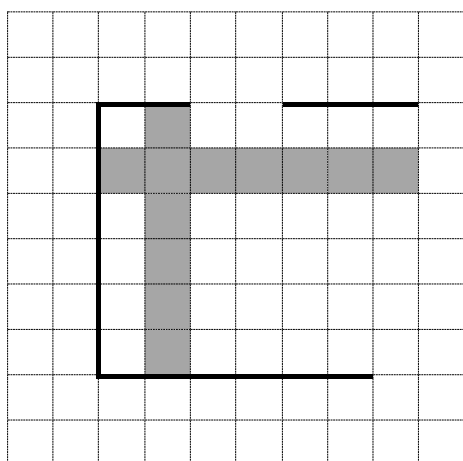
**Выполните задание.**

На бесконечном поле есть две горизонтальные стены и одна вертикальная. Левый конец первой горизонтальной стены соединён с верхним концом вертикальной стены, а нижний конец вертикальной стены соединён с левым концом второй горизонтальной стены. **Длины стен неизвестны.** В первой горизонтальной стене есть ровно один проход, точное место прохода и его ширина неизвестны. Гарантируется, что расстояние между горизонтальными стенами не менее двух клеток. Робот находится в одной из клеток, расположенных непосредственно левее вертикальной стены.

На рисунке указан один из возможных способов расположения стен и Робота (Робот обозначен буквой «Р»).



Напишите для Робота алгоритм, закрашивающий все клетки, расположенные правее вертикальной стены и отстоящие от неё ровно на одну клетку, и клетки, расположенные ниже первой горизонтальной стены, также отстоящие от неё ровно на одну клетку. Клетки под проходом, находящиеся ниже него на расстоянии одной клетки, также должны быть закрашены. Робот должен закрасить только клетки, удовлетворяющие данному условию. Например, для приведённого выше рисунка Робот должен закрасить следующие клетки (см. рисунок).



При исполнении алгоритма Робот не должен разрушиться, выполнение алгоритма должно завершиться. Конечное положение Робота может быть произвольным.

Алгоритм должен решать задачу для любого допустимого расположения стен и любого расположения и размера прохода внутри стены.

Алгоритм может быть выполнен в среде формального исполнителя или записан в текстовом редакторе.

Сохраните алгоритм в формате программы Кумир или в текстовом файле. Название файла и каталог для сохранения Вам сообщат организаторы экзамена.

**Содержание верного ответа и указания по оцениванию**  
(допускаются иные формулировки ответа, не искажающие его смысла)

Команды исполнителя будем записывать жирным шрифтом, а комментарии, поясняющие алгоритм и не являющиеся его частью, – курсивом. Начало комментария будем обозначать символом «|».

| *Поднимаемся вверх до конца вертикальной стены.*

**нц пока не справа свободно**

**вверх**

**кц**

| *Сдвигаемся в клетку у левого конца первой горизонтальной стены.*

**вправо**

| *Двигаемся вправо до прохода в стене.*

**нц пока не снизу свободно**

**вправо**

**кц**

| *Двигаемся вправо до конца прохода в стене.*

**нц пока снизу свободно**

**вправо**

**кц**

| *Двигаемся вправо до конца первой горизонтальной стены.*

**нц пока не снизу свободно**

**вправо**

**кц**

| *Сдвигаемся в клетку под правым концом первой горизонтальной стены и отходим от неё на одну клетку вниз.*

**вниз**

**влево**

**вниз**

| *Двигаясь влево до вертикальной стены, закрашиваем клетки.*

**нц пока слева свободно**

**закрасить**

**влево**

**кц**

**закрасить**

| *Сдвигаемся в клетку у верхнего конца вертикальной стены и отходим от неё вправо на одну клетку.*

**вверх**

**вправо**

| *Двигаясь вниз до второй горизонтальной стены, закрашиваем клетки.*

**нц пока снизу свободно**

**закрасить**

**вниз**

**кц**

**закрасить**

<b>Указания по оцениванию</b>	<b>Баллы</b>
Алгоритм правильно работает при всех допустимых исходных данных	2
При всех допустимых исходных данных верно следующее: 1) выполнение алгоритма завершается, и при этом Робот не разбивается; 2) закрашено не более трёх лишних клеток; 3) остались незакрашенными не более трёх клеток из числа тех, которые должны были быть закрашены	1
Задание выполнено неверно, т.е. не выполнены условия, позволяющие поставить 1 или 2 балла	0
<i>Максимальный балл</i>	2

## Вариант 5

Исполнитель Робот умеет перемещаться по лабиринту, начерченному на плоскости, разбитой на клетки. Между соседними (по сторонам) клетками может стоять стена, через которую Робот пройти не может.

У Робота есть девять команд. Четыре команды – это команды-приказы:

**вверх вниз влево вправо**

При выполнении любой из этих команд Робот перемещается на одну клетку соответственно: вверх ↑, вниз ↓, влево ←, вправо →. Если Робот получит команду передвижения сквозь стену, то он разрушится.

Также у Робота есть команда **закрасить**, при которой закрашивается клетка, в которой Робот находится в настоящий момент.

Ещё четыре команды – это команды проверки условий. Эти команды проверяют, свободен ли путь для Робота в каждом из четырёх возможных направлений:

**сверху свободно снизу свободно слева свободно справа свободно**

Эти команды можно использовать вместе с условием «если», имеющим следующий вид:

**если условие то**

*последовательность команд*

**все**

Здесь *условие* – одна из команд проверки условия.

*Последовательность команд* – это одна или несколько любых команд-приказов.

Например, для передвижения на одну клетку вправо, если справа нет стенки, и закрашивания клетки можно использовать такой алгоритм:

**если справа свободно то**

**вправо**

**закрасить**

**все**

В одном условии можно использовать несколько команд проверки условий, применяя логические связки **и**, **или**, **не**, например:

**если (справа свободно) и (не снизу свободно) то**

**вправо**

**все**

Для повторения последовательности команд можно использовать цикл «пока», имеющий следующий вид:

**нц пока условие**

*последовательность команд*

**кц**

*Например*, для движения вправо, пока это возможно, можно использовать следующий алгоритм:

**нц пока справа свободно**

**вправо**

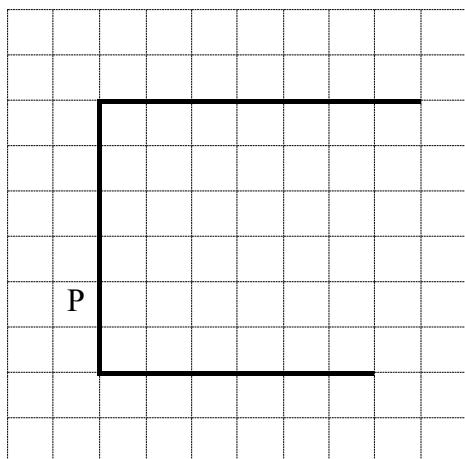
**кц**

**Выполните задание.**

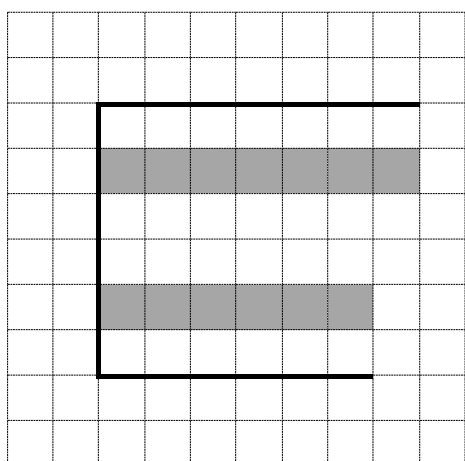
На бесконечном поле есть две горизонтальные стены и одна вертикальная. Левый конец первой горизонтальной стены соединён с верхним концом вертикальной стены, а нижний конец вертикальной стены соединён с левым концом второй горизонтальной стены. **Длины стен неизвестны.** Гарантируется, что расстояние между горизонтальными стенами

не менее двух клеток. Робот находится в одной из клеток, расположенных непосредственно левее вертикальной стены.

На рисунке указан один из возможных способов расположения стен и Робота (Робот обозначен буквой «Р»).



Напишите для Робота алгоритм, закрашивающий все клетки, расположенные над второй горизонтальной стеной и отстоящие от неё на расстоянии ровно в одну клетку, и клетки, расположенные под первой горизонтальной стеной и отстоящие от неё также на расстоянии ровно в одну клетку. Робот должен закрасить только клетки, удовлетворяющие данному условию. Например, для приведённого выше рисунка Робот должен закрасить следующие клетки (см. рисунок).



При исполнении алгоритма Робот не должен разрушиться, выполнение алгоритма должно завершиться. Конечное положение Робота может быть произвольным.

Алгоритм должен решать задачу для любого допустимого расположения стен.

Алгоритм может быть выполнен в среде формального исполнителя или записан в текстовом редакторе.

Сохраните алгоритм в формате программы Кумир или в текстовом файле. Название файла и каталог для сохранения Вам сообщат организаторы экзамена.

**Содержание верного ответа и указания по оцениванию**  
(допускаются иные формулировки ответа, не искажающие его смысла)

Команды исполнителя будем записывать жирным шрифтом, а комментарии, поясняющие алгоритм и не являющиеся его частью, – курсивом. Начало комментария будем обозначать символом «|».

*| Поднимаемся вверх до конца вертикальной стены.*

**нц пока не справа свободно**

**вверх**

**кц**

*| Сдвигаемся в клетку у левого конца первой горизонтальной стены.*

**вправо**

*| Двигаемся вправо до конца стены.*

**нц пока не снизу свободно**

**вправо**

**кц**

*| Сдвигаемся в клетку под правым концом первой горизонтальной стены и отходим от неё на одну клетку вниз.*

**вниз**

**влево**

**вниз**

*| Двигаясь влево до вертикальной стены, закрашиваем клетки.*

**нц пока слева свободно**

**закрасить**

**влево**

**кц**

**закрасить**

*| Двигаемся вниз до второй горизонтальной стены.*

**нц пока снизу свободно**

**вниз**

**кц**

*| Двигаясь вправо до конца второй горизонтальной стены, на каждом шаге отступаем от неё на одну клетку вверх, закрашиваем клетку, опускаемся на одну клетку обратно вниз.*

**нц пока не снизу свободно**

**вверх**

**закрасить**

**вниз**

**вправо**

**кц**

<b>Указания по оцениванию</b>	<b>Баллы</b>
Алгоритм правильно работает при всех допустимых исходных данных	2
При всех допустимых исходных данных верно следующее: 1) выполнение алгоритма завершается, и при этом Робот не разбивается; 2) закрашено не более трёх лишних клеток; 3) остались незакрашенными не более трёх клеток из числа тех, которые должны были быть закрашены	1
Задание выполнено неверно, т.е. не выполнены условия, позволяющие поставить 1 или 2 балла	0
<i>Максимальный балл</i>	2

## Тренировочные задания линии 16

### Вариант 1

Напишите программу подсчёта количества элементов последовательности натуральных чисел, запись которых в восьмеричной системе счисления пятизначна и оканчивается на цифру 1. В ответе запишите количество искомым элементов.

На вход программе сначала подаётся количество элементов последовательности  $N$  ( $1 \leq N \leq 1000$ ), затем – каждый элемент последовательности в отдельной строке.

Программа должна напечатать только одно число – количество искомым элементов последовательности.

*Пример работы программы*

Входные данные	Выходные данные
5 777 4353 4633 5305 32833	3

### Содержание верного ответа и указания по оцениванию

(допускаются иные формулировки ответа, не искажающие его смысла)

Решением является программа, записанная на любом языке программирования. Пример верного решения, записанного на языке Python:

```
n = int(input())
c = 0
for i in range(n):
    x = int(input())
    if (x % 8 == 1) and (4095 < x < 32768) :
        c += 1
print(c)
```

Возможны и другие варианты решения.

Для проверки правильности работы программы необходимо использовать следующие тесты.

№	Входные данные	Выходные данные
1	3 1097 4617 32777	1
2	4 2569 4227 16377 32769	1
3	3 4091 8031 19791	0

### Указания по оцениванию

### Баллы

Предложено верное решение. Программа правильно работает на всех приведённых выше тестах

2

Программа выдаёт неверный ответ только на одном из тестов, приведённых выше

1

Программа выдаёт на тестах неверные ответы, количество которых отличается от указанного в критерии на 1 балл

0

*Максимальный балл*

2

## Вариант 2

Напишите программу подсчёта количества элементов  $x_i$  последовательности натуральных чисел, таких что  $x_i > 7$  и запись  $x_i$  в восьмеричной системе счисления оканчивается на две одинаковые цифры. В ответе запишите количество искомых элементов.

Программа получает на вход натуральные числа, количество введённых чисел неизвестно, последовательность чисел заканчивается числом 0 (0 – признак окончания ввода, не входит в последовательность). Количество введённых чисел не превышает 1000.

Программа должна напечатать только одно число – количество искомых элементов последовательности.

*Пример работы программы*

Входные данные	Выходные данные
7 200 63 23322 27419 0	2

### Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)

Решением является программа, записанная на любом языке программирования. Пример верного решения, записанного на языке Python:

```
c = 0
x = int(input())
while x > 0 :
    if (x % 8 == (x // 8 ) % 8) and (x > 7) :
        c += 1
    x = int(input())
print(c)
```

Возможны и другие варианты решения.

Для проверки правильности работы программы необходимо использовать следующие тесты.

№	Входные данные	Выходные данные
1	81097 4617 27410 0	3
2	2569 4227 16377 32769 0	1
3	4091 8031 19791 0	0

Указания по оцениванию	Баллы
Предложено верное решение. Программа правильно работает на всех приведённых выше тестах	2
Программа выдаёт неверный ответ только на одном из тестов, приведённых выше	1
Программа выдаёт на тестах неверные ответы, количество которых отличается от указанного в критерии на 1 балл	0
<i>Максимальный балл</i>	2

### Вариант 3

Напишите программу подсчёта количества элементов  $x_i$  последовательности натуральных чисел, таких что  $x_i > 9$  и сумма двух последних цифр равна 11. В ответе запишите количество искоемых элементов.

Программа получает на вход натуральные числа, количество введённых чисел неизвестно, последовательность чисел заканчивается числом 0 (0 – признак окончания ввода, не входит в последовательность). Количество введённых чисел не превышает 1000.

Программа должна напечатать только одно число – количество искоемых элементов последовательности.

#### Пример работы программы

Входные данные	Выходные данные
8 200 65 74382 27429 0	2

#### Содержание верного ответа и указания по оцениванию

(допускаются иные формулировки ответа, не искажающие его смысла)

Решением является программа, записанная на любом языке программирования. Пример верного решения, записанного на языке Python:

```
c = 0
x = int(input())
while x > 0 :
    if (x % 10 + (x //10 )% 10  == 11) and (x>9) :
        c += 1
    x = int(input())
print(c)
```

Возможны и другие варианты решения.

Для проверки правильности работы программы необходимо использовать следующие тесты.

№	Входные данные	Выходные данные
1	38097 4647 27454 0	1
2	2569 4227 16377 32729 0	1
3	4065 8031 19791 0	1

Указания по оцениванию	Баллы
Предложено верное решение. Программа правильно работает на всех приведённых выше тестах	2
Программа выдаёт неверный ответ только на одном из тестов, приведённых выше	1
Программа выдаёт на тестах неверные ответы, количество которых отличается от указанного в критерии на 1 балл	0
<i>Максимальный балл</i>	2

#### Вариант 4

Напишите программу подсчёта количества элементов  $x_i$  последовательности натуральных чисел, таких что  $x_i > 9$  и предпоследняя цифра больше последней. В ответе запишите количество искомых элементов.

Программа получает на вход натуральные числа, количество введённых чисел неизвестно, последовательность чисел заканчивается числом 0 (0 – признак окончания ввода, не входит в последовательность). Количество введённых чисел не превышает 1000.

Программа должна напечатать только одно число – количество искомых элементов последовательности.

*Пример работы программы*

Входные данные	Выходные данные
8 208 43 74382 27499 0	2

#### Содержание верного ответа и указания по оцениванию

(допускаются иные формулировки ответа, не искажающие его смысла)

Решением является программа, записанная на любом языке программирования. Пример верного решения, записанного на языке Python:

```
c = 0
x = int(input())
while x > 0 :
    if (x % 10 < (x //10 )% 10 ) and (x>9) :
        c += 1
    x = int(input())
print(c)
```

Возможны и другие варианты решения.

Для проверки правильности работы программы необходимо использовать следующие тесты.

№	Входные данные	Выходные данные
1	38097 4647 27454 0	2
2	2569 4227 16377 32729 0	0

3	4065 8031 19791 0	3	
Указания по оцениванию			Баллы
Предложено верное решение. Программа правильно работает на всех приведённых выше тестах			2
Программа выдаёт неверный ответ только на одном из тестов, приведённых выше			1
Программа выдаёт на тестах неверные ответы, количество которых отличается от указанного в критерии на 1 балл			0
<i>Максимальный балл</i>			2

### Вариант 5

Напишите программу подсчёта количества элементов  $x_i$  последовательности натуральных чисел, таких что  $x_i > 9$  и предпоследняя цифра меньше последней. В ответе запишите количество искомых элементов.

Программа получает на вход натуральные числа, количество введённых чисел неизвестно, последовательность чисел заканчивается числом 0 (0 – признак окончания ввода, не входит в последовательность). Количество введённых чисел не превышает 1000.

Программа должна напечатать только одно число – количество искомых элементов последовательности.

#### Пример работы программы

Входные данные	Выходные данные
8 208 34 74382 27499 0	2

### Содержание верного ответа и указания по оцениванию

(допускаются иные формулировки ответа, не искажающие его смысла)

Решением является программа, записанная на любом языке программирования. Пример верного решения, записанного на языке Python:

```
c = 0
x = int(input())
while x > 0 :
    if (x % 10 > (x //10 )% 10 ) and (x>9) :
        c += 1
    x = int(input())
print(c)
```

Возможны и другие варианты решения.

Для проверки правильности работы программы необходимо использовать следующие тесты.

№	Входные данные	Выходные данные
1	38097 4647 27454 0	1

2	2569 4227 16377 32729 0	3	
3	4065 8031 19791 0	0	
<b>Указания по оцениванию</b>			<b>Баллы</b>
Предложено верное решение. Программа правильно работает на всех приведённых выше тестах			2
Программа выдаёт неверный ответ только на одном из тестов, приведённых выше			1
Программа выдаёт на тестах неверные ответы, количество которых отличается от указанного в критерии на 1 балл			0
<i>Максимальный балл</i>			2

### Вариант 6

Напишите программу подсчёта количества элементов  $x_i$  последовательности натуральных чисел, таких что  $x_i > 9$  и произведение двух последних цифр больше 25. В ответе запишите количество искоемых элементов.

Программа получает на вход натуральные числа, количество введённых чисел неизвестно, последовательность чисел заканчивается числом 0 (0 – признак окончания ввода, не входит в последовательность). Количество введённых чисел не превышает 1000.

Программа должна напечатать только одно число – количество искоемых элементов последовательности.

*Пример работы программы*

Входные данные	Выходные данные
8 26 155 74313 274774 0	1

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)
<p>Решением является программа, записанная на любом языке программирования. Пример верного решения, записанного на языке Python:</p> <pre> c = 0 x = int(input()) while x &gt; 0 :     if ((x % 10) * ((x // 10) % 10) &gt; 25) and (x &gt; 9) :         c += 1     x = int(input()) print(c) </pre> <p>Возможны и другие варианты решения. Для проверки правильности работы программы необходимо использовать следующие тесты.</p>

<b>№</b>	<b>Входные данные</b>	<b>Выходные данные</b>	
1	38097 4647 27454 0	2	
2	2569 4227 16377 32729 0	2	
3	4065 8031 19791 0	1	
<b>Указания по оцениванию</b>			<b>Баллы</b>
Предложено верное решение. Программа правильно работает на всех приведённых выше тестах			2
Программа выдаёт неверный ответ только на одном из тестов, приведённых выше			1
Программа выдаёт на тестах неверные ответы, количество которых отличается от указанного в критерии на 1 балл			0
<i>Максимальный балл</i>			2

**Желаем успеха на экзамене!**