

ОГЭ-2020

Д. М. Ушаков

ИНФОРМАТИКА

10

**ТРЕНИРОВОЧНЫХ ВАРИАНТОВ
ЭКЗАМЕНАЦИОННЫХ РАБОТ
ДЛЯ ПОДГОТОВКИ
К ОСНОВНОМУ ГОСУДАРСТВЕННОМУ
ЭКЗАМЕНУ**

*+ вариант
с подробным
разбором решений*

Москва
Издательство АСТ
2019

УДК 373:002
ББК 32.81я721
У93

Ушаков, Денис Михайлович.

у93 ОГЭ-2020 : Информатика : 10 тренировочных вариантов экзаменационных работ для подготовки к основному государственному экзамену / Д.М. Ушаков. — Москва: АСТ, 2019. — 138, [6] с. — (ОГЭ-2020. Это будет на экзамене).
ISBN 978-5-17-115724-1

10 вариантов экзаменационных работ по информатике — пособие для учащихся 9 классов, позволяющее в кратчайшие сроки успешно подготовиться к сдаче основного государственного экзамена.

Каждый вариант составлен в полном соответствии с требованиями государственной итоговой аттестации, включает задания разных типов и уровней сложности по основным разделам курса информатики.

Структура всех вариантов одина. Каждый из них состоит из 2-х частей и включает 20 заданий.

В пособие включён вариант с подробным разбором решений всех заданий. Представленные алгоритмы будут полезны при проверке и оценке своих навыков и умений в решении типовых экзаменационных заданий.

В конце книги даны ответы на все задания и подробный анализ заданий с развёрнутым ответом с критериями оценки.

Материалы сборника могут быть использованы для планомерного повторения изученного материала и тренировки в выполнении заданий различного типа при подготовке к экзамену.

УДК 373:002
ББК 32.81я721

ISBN 978-5-17-115724-1

© Ушаков Д.М., 2019
© ООО «Издательство АСТ», 2019

СОДЕРЖАНИЕ

Предисловие	4	Вариант 6	77
Инструкция по выполнению работы	6	Часть 1	77
ВАРИАНТ с подробным разбором решений и ответами	7	Часть 2	82
Варианты экзаменационных работ.	36	Вариант 7	85
Вариант 1	36	Часть 1	85
Часть 1	36	Часть 2	90
Часть 2	41	Вариант 8	93
Вариант 2	44	Часть 1	93
Часть 1	44	Часть 2	98
Часть 2	50	Вариант 9	101
Вариант 3	53	Часть 1	101
Часть 1	53	Часть 2	106
Часть 2	58	Вариант 10	109
Вариант 4	61	Часть 1	109
Часть 1	61	Часть 2	114
Часть 2	66	Ответы на задания. Часть 1	117
Вариант 5	69	Ответы и критерии оценки к заданиям части 2.	118
Часть 1	69	Образец экзаменационных бланков	139
Часть 2	74		

ПРЕДИСЛОВИЕ

Уважаемые учащиеся 9-х классов, абитуриенты и учителя!

Вашему вниманию предлагается сборник тренировочных вариантов экзаменационных работ по информатике для подготовки к ОГЭ в 2020 году.

Содержание заданий разработано по основным темам курса информатики и ИКТ, объединенным в следующие тематические блоки: «Представление и передача информации», «Обработка информации», «Основные устройства ИКТ», «Запись средствами ИКТ информации об объектах и о процессах, создание и обработка информационных объектов», «Проектирование и моделирование», «Математические инструменты, электронные таблицы», «Организация информационной среды, поиск информации».

В начале пособия включён вариант с подробным разбором решений всех заданий. Представленные алгоритмы будут полезны при проверке и оценке своих навыков и умений в решении типовых экзаменационных заданий. Этот материал будет «палочкой-выручалочкой» при возникших сложностях в работе с тем или иным заданием!

Данный сборник содержит 10 типовых вариантов экзаменационных работ, составленных в соответствии с демонстрационным вариантом и спецификацией 2019 года.

Каждый вариант состоит из двух частей и включает в себя 20 заданий, различающихся формой и уровнем сложности.

Часть 1 содержит 18 заданий с кратким ответом нескольких разновидностей:

- задания на выбор и запись одного или нескольких правильных ответов из предложенного перечня ответов;
- задания на вычисление определённой величины;
- задания на установление правильной последовательности, представленной в виде строки символов по определённому алгоритму.

В части 1 11 заданий относятся к базовому уровню и 7 заданий к повышенному уровню сложности.

Задания проверяют материал всех тематических блоков.

Часть 2 содержит 2 задания высокого уровня сложности. При этом одно из этих заданий (номер 20) предлагает выбор одного из двух вариантов.

Задания части 1 выполняются экзаменуемыми без использования компьютеров и других технических средств. Вычислительная сложность заданий не требует использования калькуляторов, поэтому в целях обеспечения равенства всех участников экзамена использование калькуляторов на экзаменах не разрешается.

Задания части 2 выполняются экзаменуемыми на компьютере. На компьютере должны быть установлены знакомые обучающимся программы.

Для выполнения задания 19 необходима программа для работы с электронными таблицами.

Задание 20 (на составление алгоритма) даётся в двух вариантах по выбору обучающегося. Первый вариант задания (20.1) предусматривает разработку алгоритма для исполнителя «Робот». Для выполнения задания 20.1 рекомендуется использование учебной среды исполнителя «Робот». В качестве такой среды может использоваться, например, учебная среда разработки «Кумир», разработанная в НИИСИ РАН (<http://www.niisi.ru/kumir>), или любая другая среда, позволяющая моделировать исполнителя «Робот».

В случае, если синтаксис команд исполнителя используемой среде отличается от того, который дан в задании, допускается внесение изменений в текст задания в части описания исполнителя «Робот». При отсутствии учебной среды исполнителя «Робот» решение задания 20.1 записывается в простом текстовом редакторе.

Второй вариант задания (20.2) предусматривает запись алгоритма на изучаемом языке программирования (если изучение темы «Алгоритмизация» проводится с использованием языка программирования). В этом случае для выполнения задания необходима система программирования, используемая при обучении.

Выполнением каждого задания части 2 является отдельный файл, подготовленный в соответствующей программе (текстовом редакторе или электронной таблице).

В конце пособия представлены ответы для проверки решений.

Если при решении заданий из данного пособия Вы обнаружите какие-то неточности или опечатки, то на странице авторского сайта www.dmushakov.ru можно посмотреть самые последние сведения о замеченных ошибках. Там же можно задать вопрос автору сборника и посмотреть, какие дополнительные пособия Д. М. Ушакова по информатике, издаваемые в нашем издательстве, могут быть Вам полезны при подготовке к экзамену.

В связи с возможными изменениями в формате и количестве заданий рекомендуем в процессе подготовки к экзамену обращаться к материалам сайта официального разработчика экзаменационных заданий — Федерального института педагогических измерений: www.fipi.ru.

ИНСТРУКЦИЯ ПО ВЫПОЛНЕНИЮ РАБОТЫ

Экзаменационная работа состоит из двух частей, включающих в себя 20 заданий. Часть 1 содержит 18 заданий с кратким ответом, часть 2 содержит 2 задания, которые необходимо выполнить на компьютере.

На выполнение экзаменационной работы по информатике отводится 2 часа 30 минут (150 минут). К выполнению заданий части 2 можно перейти, только сдав выполненные задания части 1 экзаменационной работы. Вы можете самостоятельно определять время, которое отводите на выполнение заданий части 1, но рекомендуемое время — 1 час 15 минут (75 минут) и на выполнение заданий части 2 также 1 час 15 минут (75 минут).

При выполнении заданий части 1 **нельзя** пользоваться компьютером, калькулятором, справочной литературой.

Ответы к заданиям 1–6 записываются в виде одной цифры, которая соответствует номеру правильного ответа. Эту цифру запишите в поле ответа в тексте работы.

Ответы к заданиям 7–18 записываются в виде числа, последовательности цифр или букв в поле ответа в тексте работы.

В случае записи неверного ответа на задания части 1 зачеркните его и запишите рядом новый.

Часть 2 содержит 2 задания (19, 20). Результатом выполнения каждого из этих заданий является отдельный файл. Формат файла, его имя и каталог для сохранения Вам сообщат организаторы экзамена.

При выполнении заданий можно пользоваться черновиком. **Записи в черновике, а также в тексте контрольных измерительных материалов не учитываются при оценивании работы.**

Баллы, полученные Вами за выполненные задания, суммируются. Постарайтесь выполнить как можно больше заданий и набрать наибольшее количество баллов.

После завершения работы проверьте, чтобы ответ на каждое задание в бланке ответов № 1 был записан под правильным номером.

Желаем успеха!

ВАРИАНТ

с подробным разбором решений и ответами

Часть 1

Ответом к заданиям 1–6 является одна цифра, которая соответствует номеру правильного ответа. Запишите эту цифру в поле ответа в тексте работы, а затем перенесите в БЛАНК ОТВЕТОВ № 1 справа от номера соответствующего задания, начиная с первой клеточки.

1 В одной из кодировок Unicode каждый символ кодируется двумя байтами. Определите информационный объём слова из тридцати двух символов в этой кодировке.

- 1) 512 байт 2) 32 бит 3) 512 бит 4) 32 байта

Ответ: _____.

Решение.

Информационный объём сообщения — это общее количество информации в сообщении. Так как по условию каждый символ кодируется двумя байтами, чтобы найти объём всего сообщения, нужно умножить количество символов в сообщении на количество информации в одном символе. Получаем $32 \cdot 2$ байта = 64 байта. Ищем такой вариант ответа. Не находим. Однако, видим среди предлагаемых вариантов ответа количество информации в битах. Переведём полученные 64 байта в биты. Для этого умножим их на 8 (количество бит в одном байте). Получаем: $64 \cdot 8 = 512$ бит. Ищем такой вариант среди ответов.

О т в е т: 3.

Другой вариант задания 1: «Дана фраза и количество информации в одном символе. Нужно определить информационный объём сообщения.»

В этом случае задача решается так же — нужно умножить количество символов на количество информации в одном символе. То есть, нужно аккуратно посчитать количество символов в сообщении и умножить на данное количество информации в одном символе. Очень важно при этом правильно подсчитывать количество символов. Символами сообщения при этом будут являться не только видимые символы, но также знаки препинания и пробелы. Не забудьте их при подсчёте количества символов в сообщении.

2 Для какого из приведённых чисел ложно высказывание:

НЕ (число < 20) **ИЛИ** (число нечётное)?

- 1) 18 2) 48 3) 35 4) 13

Ответ: _____.

Решение.

Способ 1.

Выполним по действиям для каждого приведённого в вариантах ответа числа указанные операции. Для этого, для начала, определим, в каком порядке нужно будет выполнять приведённые операции. Сначала нужно выполнить выражения в скобках. В данном случае, их два. Выполняем их слева направо. После этого нужно определить, в каком порядке выполнять операции **НЕ** и **ИЛИ**. Для этого вспомним, что приоритет логических операций: **НЕ-И-ИЛИ**. Значит, в нашем случае сначала выполнится операция **НЕ**, а затем **ИЛИ**. Таким образом, порядок выполнения операций будет таким:

3 1 4 2
НЕ (число < 20) **ИЛИ** (число нечётное)

Для удобства записи этих действий оформим таблицу:

Номер операции	1	2	3	4
Число	Число < 20	Число нечётное	НЕ (1)	(3) ИЛИ (2)
18	1	0	0	0
48	0	0	1	1
35	0	1	1	1
13	1	1	0	1

Находим в последнем столбце таблицы единственный 0 — он соответствует числу (18), для которого данное логическое высказывание ложно.

О т в е т: 1.

Способ 2.

Анализируем данное логическое выражение. Последняя операция, которая выполняется в выражении — ИЛИ. Так как его результат должен быть ложным, каждый операнд должен быть ложным. То есть, НЕ (число < 20) должно быть ложным, И (число нечётное) должно быть ложным. Для этого (число < 20) должно быть истинным и, чтобы (число нечётное) было ложным, его противоположность (число чётное) должно быть истинным. Получаем, что требуемое число должно быть меньше 20-ти и при этом чётным. Ищем такое число среди вариантов ответа. Подходит только число 18 (вариант ответа 1).

О т в е т: 1.

3

Между населёнными пунктами А, В, С, D, Е построены дороги, протяжённость которых (в километрах) приведена в таблице. (Отсутствие числа в таблице означает, что прямой дороги между пунктами нет.)

	А	В	С	D	Е
А		2	5	1	
В	2		1	2	
С	5	1		3	2
D	1	2	3		
Е			2		

Определите длину кратчайшего пути между пунктами А и Е (при условии, что передвигаться можно только по построенным дорогам).

1) 4

2) 5

3) 6

4) 7

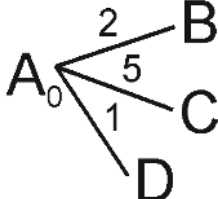
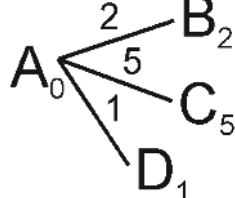
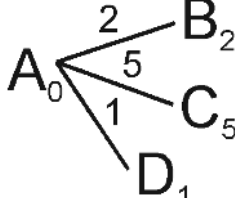
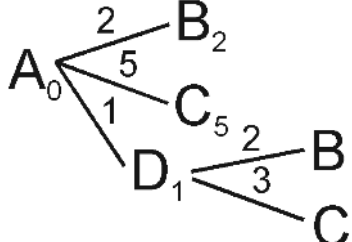
Ответ: _____.

Р е ш е н и е.

Для решения этой задачи воспользуемся специальным алгоритмом, который эффективно решает именно такого вида задачу — находит кратчайшее расстояние на графе от одной вершины до другой. Этот алгоритм называется *алгоритмом Дейкстры*. Побочным эффектом алгоритма Дейкстры является нахождение кратчайшего расстояния от некоей стартовой вершины до всех остальных вершин графа. Но это никоим образом не ухудшает его качества — алгоритм Дейкстры ищет кратчайшее расстояние быстро, надёжно и эффективно.

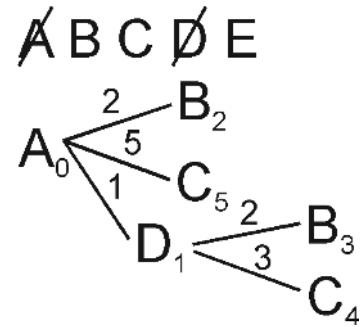
0. Выпишем в отдельный список все возможные вершины графа.

А В С D Е

<p>1. Будем строить дерево. Нарисуем стартовую вершину (в нашем случае это точка A) и подпишем рядом с ней расстояние, которое нужно «проехать», чтобы добраться до нее из стартовой точки A. То есть, ноль. Эта вершина будет корнем нашего дерева.</p>	<p>A B C D E</p> <p>A_0</p>
<p>2. Из всех возможных листовых вершин дерева найдём вершину с наименьшим числом (в данном случае, такая вершина только одна — A). Вычеркнем эту вершину из списка доступных вершин. В дальнейшем её мы больше не будем рассматривать в качестве возможной вершины дерева. То есть, вычеркнутые вершины в дерево больше не дорисовываются! Расстояния до них от стартовой вершины подсчитаны окончательно и лучше уже не будет. Следующее действие будем делать из этой вершины.</p>	<p>A B C D E</p> <p>A_0</p>
<p>3. По таблице расстояний найдём все вершины, до которых идёт ребро из текущей вершины (сейчас это вершина A) и которые ещё не вычеркнуты. Это вершины B, C, D. Нарисуем из текущей вершины (A) ветви дерева для каждой из найденных смежных вершин, нарисуем на концах ветвей названия этих вершин. Подпишем на ветвях длины ребёр по таблице расстояний.</p>	<p>A B C D E</p> 
<p>4. Для каждой полученной вершины посчитаем расстояние до неё как: расстояние до текущей вершины (подписано рядом с ней, сейчас для вершины A это 0) плюс расстояние (длины ребра) до каждой полученной вершины (подписано над ребром). Запишем эти расстояния рядом с каждой полученной вершиной.</p>	<p>A B C D E</p> 
<p>5. Среди листовых вершин найдём вершину с наименьшим расстоянием. В данном случае это вершина D. Вычеркнем эту вершину из списка доступных вершин. Дальнейшие действия будем делать из неё. Заметим, это такое же действие, как действие 2.</p>	<p>A B C D E</p> 
<p>6. По таблице расстояний найдём все вершины, до которых идёт ребро из текущей вершины (сейчас это вершина D) и которые ещё не вычеркнуты. Это вершины B и C. Нарисуем из текущей вершины (D) ветви дерева для каждой из найденных смежных вершин, нарисуем на концах ветвей названия этих вершин. Подпишем на ветвях длины ребёр по таблице расстояний. Заметим, это такое же действие, как действие 3.</p>	<p>A B C D E</p> 

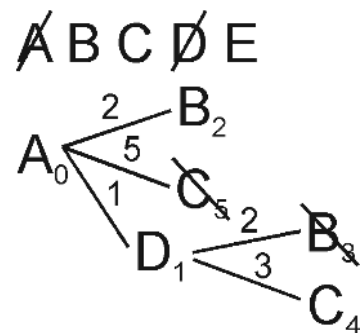
7. Для каждой полученной вершины посчитаем расстояние до неё как: расстояние до текущей вершины (подписано рядом с ней, сейчас для вершины **D** это 1) плюс расстояние (длины ребра) до каждой полученной вершины (подписано над ребром). Запишем эти расстояния рядом с каждой полученной вершиной.

Заметим, это такое же действие, как действие 4.

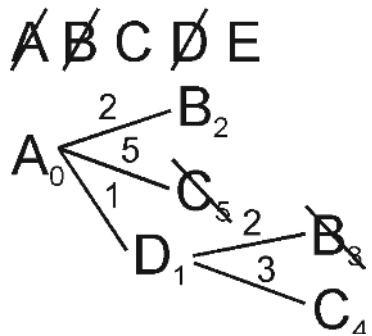


8. Среди всех листовых вершин будем искать одинаковые вершины. Сейчас это пара вершин B_2 и B_3 и пара вершин C_5 и C_4 . Для каждой пары одинаковых вершин будем вычеркивать в дереве худшую вершину. То есть, вершину с бóльшим расстоянием до неё. Сейчас это B_3 и C_5 . Если в паре одинаковых вершин расстояния будут одинаковые, нужно вычеркнуть одну любую. Мы специально вычеркиваем вершины указанным образом (от левого верхнего к правому нижнему углу), чтобы перечеркнуть как название вершины, так и расстояние до неё. Это облегчит в дальнейшем поиск вершины с наименьшим расстоянием.

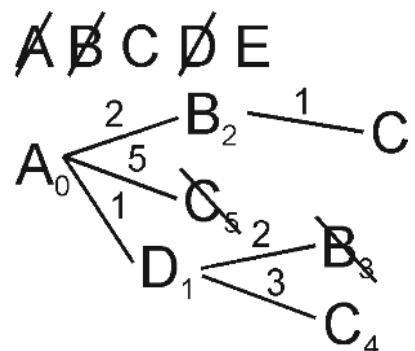
Заметим, что этого действия (вычеркивания) мы раньше не делали, потому что у нас ещё не встречались одинаковые вершины. В действительности, это часть циклического алгоритма. Действия 5–8 мы будем повторять, пока не вычеркнем конечную вершину из списка доступных вершин.



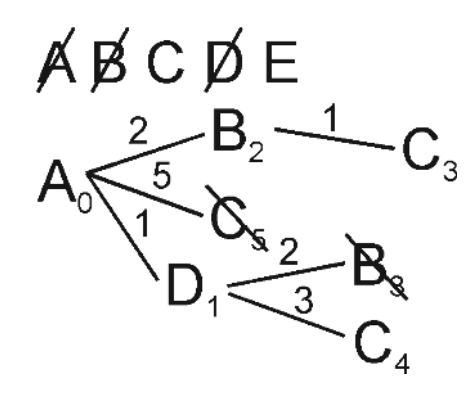
9. Среди листовых вершин найдём вершину с наименьшим расстоянием. В данном случае это вершина **B**. Вычеркнем эту вершину из списка доступных вершин. Дальнейшие действия будем делать из неё.



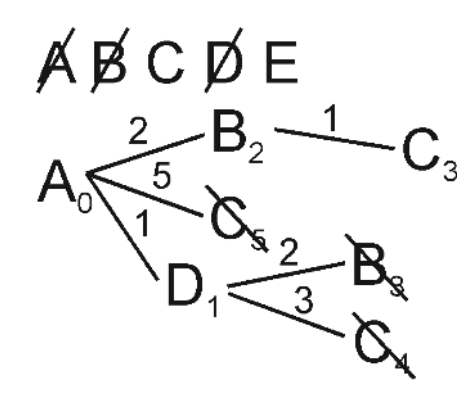
10. По таблице расстояний найдём все вершины, до которых идёт ребро из текущей вершины (сейчас это вершина **B**) и которые ещё не вычеркнуты. Это вершина **C**. Нарисуем из текущей вершины (**B**) ветви дерева для каждой из найденных смежных вершин, нарисуем на концах ветвей названия этих вершин. Подпишем на ветвях длины рёбер по таблице расстояний.



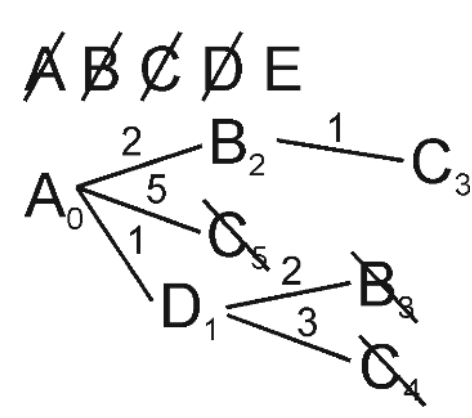
11. Для каждой полученной вершины посчитаем расстояние до неё как: расстояние до текущей вершины (подписано рядом с ней, сейчас для вершины В это 2) плюс расстояние (длины ребра) до каждой полученной вершины (подписано над ребром). Запишем эти расстояния рядом с каждой полученной вершиной. Сейчас это C_3 .



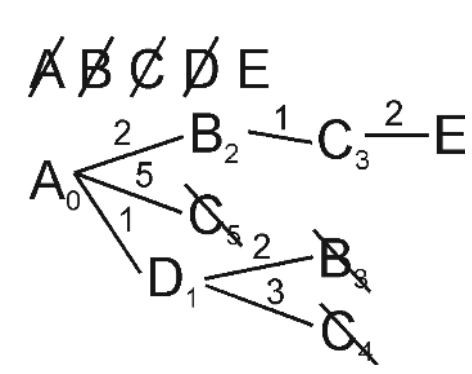
12. Среди всех листовых вершин будем искать одинаковые вершины. Сейчас это C_3 и C_4 . Для каждой пары одинаковых вершин будем вычеркивать в дереве худшую вершину. То есть, вершину с бóльшим расстоянием до неё. Сейчас это C_4 .



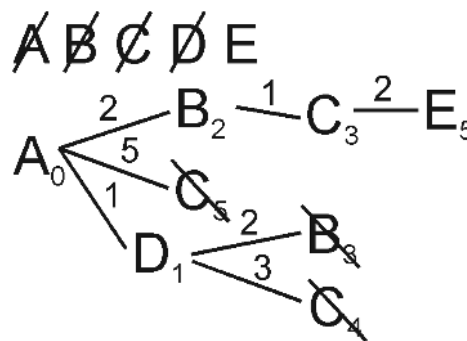
13. Среди листовых вершин найдём вершину с наименьшим расстоянием. В данном случае это вершина С. Вычеркнем эту вершину из списка доступных вершин. Дальнейшие действия будем делать из неё.



14. По таблице расстояний найдём все вершины, до которых идёт ребро из текущей вершины (сейчас это вершина С) и которые ещё не вычеркнуты. Это вершина Е. Нарисуем из текущей вершины (С) ветви дерева для каждой из найденных смежных вершин, на концах ветвей названия этих вершин. Подпишем на ветвях длины ребер по таблице расстояний.



15. Для каждой полученной вершины посчитаем расстояние до неё как: расстояние до текущей вершины (подписано рядом с ней, сейчас для вершины С это 3) плюс расстояние (длины ребра) до каждой полученной вершины (подписано над ребром). Запишем эти расстояния рядом с каждой полученной вершиной. Сейчас это E_5 .



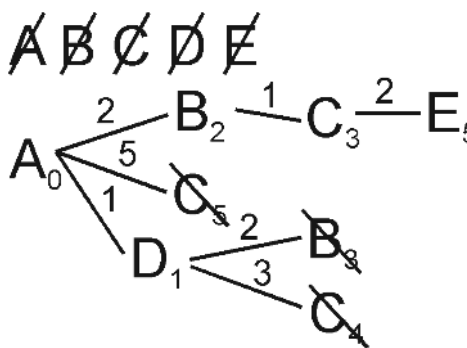
16–17. Среди всех листовых вершин будем искать одинаковые вершины. Сейчас таких пар нет.

Среди листовых вершин найдем вершину с наименьшим расстоянием. В данном случае это вершина E. Вычеркнем эту вершину из списка доступных вершин.

Это конечная вершина, расстояние до которой мы искали. Алгоритм закончен.

Кратчайшее расстояние: 5.

О т в е т : 2.



4

В некотором каталоге хранился файл **Колобок.jpg**, имевший полное имя **C:\Сказки\Герои\Колобок.jpg**. Пользователь, находившийся в этом каталоге, поднялся на один уровень вверх, создал подкаталог **Главные**, переместился в этот подкаталог, создал в нём подкаталог **Хорошие** и переместил в созданный подкаталог весь каталог **Герои**. Каково стало полное имя файла **Колобок.jpg** после перемещения?

- 1) C:\Главные\Хорошие\Герои\Колобок.jpg
- 2) C:\Сказки\Главные\Хорошие\Герои\Колобок.jpg
- 3) C:\Сказки\Главные\Хорошие\Колобок.jpg
- 4) C:\Главные\Хорошие\Герои\Колобок.jpg

Ответ: _____.

Р е ш е н и е.

Разберём аккуратно по действиям всё, что сделал пользователь.

Так как полное имя файла было **C:\Сказки\Герои\Колобок.jpg**, делаем вывод, что пользователь изначально находился в каталоге **C:\Сказки\Герои**. Он поднялся на один уровень вверх. Значит, он оказался в каталоге **C:\Сказки**. В этом каталоге он создал подкаталог **Главные** и переместился в него. Значит, он оказался в каталоге **C:\Сказки\Главные**. В этом каталоге он создал подкаталог **Хорошие**. Значит, полное имя этого подкаталога: **C:\Сказки\Главные\Хорошие**. В этот каталог пользователь переместил весь каталог **Герои** (то есть, вместе с файлом **Колобок.jpg**). Значит, полное имя файла **Колобок.jpg** стало после этого: **C:\Сказки\Главные\Хорошие\Герои\Колобок.jpg**. Ищем такой вариант среди ответов.

О т в е т : 2.

5 Дан фрагмент электронной таблицы.

	A	B	C	D
1	7	3	4	1
2	=A1-B1	=C1*2		=B1-D1

Какая из формул, приведённых ниже, может быть записана в ячейке C2, чтобы построенная после выполнения вычислений диаграмма по значениям диапазона ячеек A2:D2 соответствовала рисунку?



- 1) =C1-B1 2) =A1-B1 3) =D1*2 4) =A1-2*D1

Ответ: _____.

Решение.

Вычислим значение всех формул, записанных в строке 2:

$$A2 = A1 - B1 = 7 - 3 = 4$$

$$B2 = C1 * 2 = 4 * 2 = 8$$

$$D2 = B1 - D1 = 3 - 1 = 2$$

Подставим эти значения в таблицу:

	A	B	C	D
2	4	8		2

Анализируем круговую диаграмму. Соотношение секторов на этой диаграмме показывает соотношение чисел в ячейках, по которым построена диаграмма. Самые маленькие секторы на диаграмме составляют 1/8 часть круга. Они в 2 раза меньше, чем средний по величине сектор круга (он составляет 1/4 часть круга). Также они в 4 раза меньше, чем самый большой сектор круга (он составляет 1/2 часть круга). Получаем, что числа в ячейках A2:D2 должны находиться в соотношении: 1:1:2:4. Имеющиеся у нас значения: 4, 8, 2. Сопоставим эти значения требуемым коэффициентам в соотношении. Самое маленькое число — 2. Среднее — 4. Самое большое — 8. Они относятся друг к другу как 1:2:4. Значит, не хватает ещё одного самого маленького числа. То есть, числа 2.

Вычислим формулы, которые предлагают нам варианты ответа:

$$1) = C1 - B1 = 4 - 3 = 1$$

$$2) = A1 - B1 = 7 - 3 = 4$$

$$3) = D1 * 2 = 1 * 2 = 2$$

$$4) = A1 - 2 * D1 = 7 - 2 * 1 = 5$$

Найдём среди них ту, результат которой равен 2.

О т в е т : 3.

6 Исполнитель Чертёжник перемещается на координатной плоскости, оставляя след в виде линии. Чертёжник может выполнять команду **Сместиться на (a, b)** (где a, b — целые числа), перемещающую Чертёжника из точки с координатами (x, y) в точку с координатами (x + a, y + b). Если числа a, b положительные, значение соответствующей координаты увеличивается; если отрицательные — уменьшается.

Например, если Чертёжник находится в точке с координатами $(9, 5)$, то команда **Сместиться на $(1, -2)$** переместит Чертёжника в точку $(10, 3)$.

Запись

Повтори k раз

Команда1 Команда2 Команда3

конец

означает, что последовательность команд **Команда1 Команда2 Команда3** повторится k раз.

Чертежнику был дан для исполнения следующий алгоритм:

Повтори 5 раз

Сместиться на $(-2, -4)$ Сместиться на $(1, 3)$ Сместиться на $(-3, 2)$

конец

На какую одну команду можно заменить этот алгоритм, чтобы Чертёжник оказался в той же точке, что и после выполнения алгоритма?

- 1) Сместиться на $(4, -1)$
- 2) Сместиться на $(20, -5)$
- 3) Сместиться на $(-4, 1)$
- 4) Сместиться на $(-20, 5)$

Ответ: _____.

Решение.

Перемещение Чертёжника по оси x никак не влияет на его перемещение по оси y . Будем производить все действия отдельно для оси x и отдельно — для оси y . Сначала определим общее смещение Чертёжника за один шаг цикла.

По оси x : $-2 + 1 - 3 = -4$.

По оси y : $-4 + 3 + 2 = 1$.

Так как цикл выполняется 5 раз, общее смещение по каждой оси нужно умножить на 5:

По оси x : $-4 \cdot 5 = -20$.

По оси y : $1 \cdot 5 = 5$.

Требуется определить команду, которая выполнит то же, что и приведенный алгоритм. Это команда **Сместиться на $(-20, 5)$** . Ищем такую команду среди вариантов ответа.

О т в е т: 4.

Ответами к заданиям 7–18 являются число, последовательность букв или цифр, которые следует записать в БЛАНК ОТВЕТОВ № 1 справа от номера соответствующего задания, начиная с первой клеточки, без пробелов, запятых и других дополнительных символов. Каждый символ пишете в отдельной клеточке в соответствии с приведёнными в бланке образцами.

7

Разведчик передал в штаб радиogramму

• - - - • • • • - - - - • - - - •

В этой радиogramме содержится последовательность букв, в которой встречаются только буквы А, Б, В, Г, Д. Каждая буква закодирована с помощью азбуки Морзе. Разделителей между кодами букв нет. Запишите в ответе переданную последовательность букв.

Нужный фрагмент азбуки Морзе приведён ниже.

А	Б	В	Г	Д
- •	- - •	• •	• -	- - -

Ответ: _____.