

Предисловие

Уважаемые читатели! Вы держите в руках второй сборник контрольных тренировочных материалов для подготовки к Единому государственному экзамену (ЕГЭ) по информатике и ИКТ, первый вышел годом ранее. Сборник предназначен как для учеников 10–11-х классов средних школ, так и для их учителей.

Настоящее пособие призвано помочь систематизировать изученный материал, а также выявить и устранить пробелы в знаниях.

В настоящее время в России изданы учебники разных авторов по информатике и ИКТ для старших классов средних общеобразовательных и специализированных школ. Последовательность изложения материала и его объём в разных учебниках зачастую существенно различаются. Соответственно различаются и учебные программы по информатике и ИКТ в разных школах. В одних учебниках делается упор на моделирование, в других — на программирование, в третьих — на информационные технологии и т.п. Поэтому научиться решать задания ЕГЭ, имея только один учебник по информатике и ИКТ, очень сложно.

В сборнике приводятся подробные решения одного примерного варианта ЕГЭ по информатике и пять вариантов для самостоятельного решения с ответами и комментариями.

Все задания снабжены ссылками на теоретический материал и примеры в учебно-справочных материалах для подготовки к ЕГЭ¹.

Для подготовки к экзамену и решения некоторых заданий рекомендуем использовать свободно распространяемые системы программирования, разработанные Научно-исследовательским институтом системных исследований Российской академии наук (НИИСИ РАН):

- КуМир (<http://www.niisi.ru/kumir-ege.htm>) — среда программирования на школьном алгоритмическом языке (школьном АЯ);
- FreePascal (<http://www.niisi.ru/fp.htm>) — среда программирования на языке Паскаль.

¹ Информатика и ИКТ: ЕГЭ: Учебно-справочные материалы для 11 класса (Серия «Итоговый контроль: ЕГЭ») / С. М. Авдошин, Р. З. Ахметсафина, О. В. Максименкова. — М.; СПб.: Просвещение, 2012.

Авторы пособия работают в Национальном исследовательском университете «Высшая школа экономики» на отделении программной инженерии. Все авторы имеют опыт разработки олимпиадных и тестовых заданий по информатике для выпускников школ и студентов. В сборнике предложены оригинальные решения некоторых заданий ЕГЭ.

Мы благодарим за помощь в составлении и проверке контрольных тренировочных материалов наших студентов: Лидию Антонову, Людмилу Галкину, Елену Кальсину, Елену Козлову, Дениса Кильдишева, Александру Малахову, Ксению Сергееву, Даниила Ширгазина.

Авторы выражают глубокую признательность Галине Васильевне Этиной, заместителю директора по информационным технологиям лицея № 126 Санкт-Петербурга, за большую помощь в работе над сборником, доброжелательную и конструктивную критику.

Надеемся, что материал сборника окажется полезным для вас и поможет успешно сдать Единый государственный экзамен по информатике и информационно-коммуникационным технологиям.

Ваши вопросы, замечания, пожелания, решения заданий вы можете присылать на адреса авторов, которые вы найдёте на сайте Национального исследовательского университета «Высшая школа экономики» www.hse.ru

Авторы также приглашают всех заинтересованных читателей в группы в социальных сетях:

<http://www.facebook.com/groups/gia.ege/> и

<http://vkontakte.ru/club30019188>

Успехов и удачи!

Инструкция по выполнению работы

Экзаменационная работа состоит из трёх частей, содержащих 32 задания.

Часть 1 содержит 14 заданий с выбором ответа. К каждому заданию даётся 4 варианта ответа, из которых только один правильный.

Часть 2 состоит из 14 заданий с кратким ответом (к этим заданиям вы должны самостоятельно сформулировать и записать ответ).

Часть 3 состоит из 4 заданий. Для выполнения заданий этой части вам необходимо написать развёрнутый ответ в произвольной форме.

На выполнение частей 1 и 2 рекомендуется отводить 1,5 часа (90 минут). На выполнение заданий части 3 — 2,5 часа (150 минут).

В заданиях используются следующие *соглашения*:

1. Обозначения для логических связей (операций):

- *отрицание* (инверсия, логическое НЕ) обозначается \neg (например, $\neg A$);
- *конъюнкция* (логическое умножение, логическое И) обозначается \wedge (например, $A \wedge B$) либо $\&$ (например, $A \& B$);
- *дизъюнкция* (логическое сложение, логическое ИЛИ) обозначается \vee (например, $A \vee B$);
- *следование* (импликация) обозначается \rightarrow (например, $A \rightarrow B$);
- *строгая дизъюнкция* (исключающее ИЛИ, сложение по модулю 2) обозначается \oplus (например, $A \oplus B$). Результатом операции строгой дизъюнкции является ИСТИНА, если значения A и B различны;
- *эквиваленция* (равносильность, тождество) обозначается \equiv либо \leftrightarrow (например, $A \equiv B$, $A \leftrightarrow B$). Выражение $A \equiv B$ истинно тогда и только тогда, когда значения A и B совпадают (либо оба истинны, либо оба ложны);
- символ 1 используется для обозначения истины (истинного высказывания); символ 0 — для обозначения лжи (ложного высказывания).

2. Обозначения для множеств:
 - объединение множеств обозначается \cup (например, $A \cup B$);
 - пересечение множеств обозначается \cap (например, $A \cap B$);
 - мощность множества обозначается $| \cdot |$ (например, $|A|$).
3. Два логических выражения, содержащих переменные, называются равносильными (эквивалентными), если значения этих выражений совпадают при любых значениях переменных. Так, выражения $A \rightarrow B$ и $(\neg A) \vee B$ равносильны, а $A \vee B$ и $A \wedge B$ — нет (значения выражений разные, например, при $A = 1, B = 0$).
4. Приоритеты логических операций в порядке убывания:
 - 1) инверсия (отрицание);
 - 2) конъюнкция (логическое умножение);
 - 3) дизъюнкция (логическое сложение), строгая дизъюнкция (исключающее ИЛИ);
 - 4) импликация (следование), эквиваленция (равносильность, тождество).

Таким образом, $\neg A \wedge B \vee C \wedge D$ означает то же, что и $((\neg A) \wedge B) \vee (C \wedge D)$.

Возможна запись $A \wedge B \wedge C$ вместо $(A \wedge B) \wedge C$.

То же относится и к дизъюнкции: возможна запись $A \vee B \vee C$ вместо $(A \vee B) \vee C$.

Контрольные тренировочные задания

Вариант 1 (с решениями)

Часть 1

При выполнении заданий части 1 в бланке ответов № 1 под номером выполняемого вами задания (A1–A14) поставьте знак «X» в клеточке, номер которой соответствует номеру выбранного вами ответа.

A1 Сколько значащих нулей в двоичной записи числа 1156?

- 1) 7 2) 8 3) 9 4) 6

Решение. Переведём число 1156 в двоичную систему счисления. Для того чтобы сделать это быстро, выделим максимальные степени двойки из числа, затем из оставшейся части числа и т. д.

$$1156 - 1024 = 132;$$

$$132 - 128 = 4.$$

$$\text{Итак, } 1156 = 1024 + 128 + 4 = 2^{10} + 2^7 + 2^2.$$

Десятый, седьмой и второй разряды записи двоичного числа заполним единицами, остальные — нулями, получим: $1156_{10} = 10010000100_2$. (Нумерация разрядов в записи целого числа начинается с нуля и увеличивается справа налево.) Количество значащих нулей равно 8.

Заметим, что не обязательно записывать число в двоичной системе счисления, чтобы определить число значащих нулей. Выделив степени двойки, можно определить число разрядов и число единиц в двоичной записи числа. Этого достаточно, чтобы вычислить количество значащих нулей. Так, в данном примере общее количество разрядов двоичной записи числа равно 11 ($1024 = 2^{10} < 1156 < 2^{11}$), количество единиц равно 3. Количество значащих нулей равно $11 - 3 = 8$.

Ответ: 2.

См. учебно-справочные материалы: Перевод числа из одной позиционной системы счисления в другую, с. 44.

A2 Между населёнными пунктами A, B, C, D, E, F построены дороги, протяжённость которых приведена в таблице.

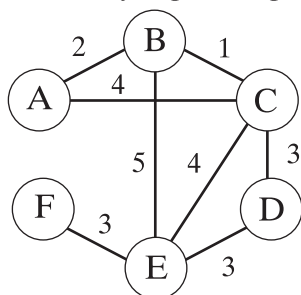
(Отсутствие числа в таблице означает, что прямой дороги между пунктами нет.)

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

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

- 1) 9 2) 10 3) 11 4) 12

Решение. Для решения задачи представим населённые пункты и расстояния между ними в виде графа. Вершины графа соответствуют населённым пунктам, они обозначены кружками, в которых записаны их обозначения. Дороги между ними обозначены линиями (рёбрами графа). Пометим рёбра числами-расстояниями между парами вершин. Получим:



Для решения воспользуемся алгоритмом Дейкстры поиска кратчайшего расстояния от одной вершины графа до других. Определим понятия:

- метка вершины X — расстояние (длина пути) от вершины А до вершины X. В решении длину пути будем обозначать AX;

- посещённая вершина — вершина, для которой определено минимальное расстояние от вершины А;
- соседи вершины X — все вершины, в которые ведут рёбра из вершины X.

Алгоритм работает пошагово — на каждом шаге «посещается» одна вершина и делается попытка уменьшить значения меток. Посещённые вершины на следующих шагах алгоритма не рассматриваются. Алгоритм Дейкстры останавливается, когда все вершины помечены как посещённые.

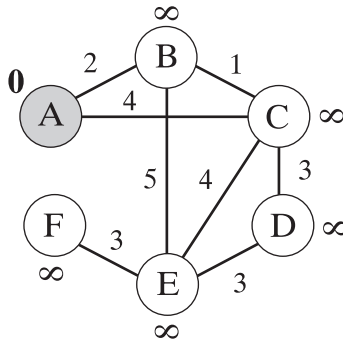
Шаг 0. Начальной вершине А присвоим метку 0, всем остальным вершинам — метку ∞ . Все вершины пока непосещённые.

Шаг 1. Из непосещённых вершин выбираем вершину V с минимальной меткой. Для каждого соседа вершины V, кроме отмеченных как посещённые, определим новую длину пути, равную сумме значений текущей метки V и длины ребра, соединяющего V с этим соседом. Если полученное значение длины меньше значения метки соседа, заменим значение метки полученным значением длины. Рассмотрев всех соседей, пометим вершину V как посещённую.

Шаг 2. Если вершина F помечена как посещённая, завершим работу алгоритма, иначе — выполним Шаг 1.

Выполним алгоритм. Будем записывать метки вершин и помечать посещённые вершины зачёркиванием.

Шаг 0. Пометим вершины. Метки запишем рядом с вершинами.

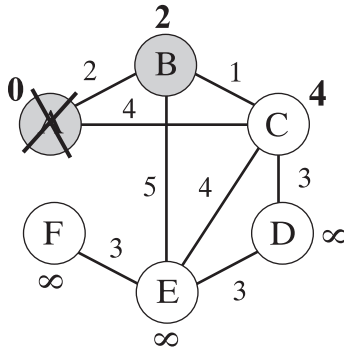


Шаг 1-1. Выберем вершину с минимальной меткой — это вершина А (она выделена цветом). Определим метки соседей, сложив метку вершины А с расстоянием до соседней вершины. Если полученное значение меньше метки соседней вершины, запишем его как метку соседней вершины:

$$AB = 0 + 2 = 2; 2 < \infty;$$

$$AC = 0 + 4 = 4; 4 < \infty.$$

Пометим вершину A как посещённую (зачёркиванием).

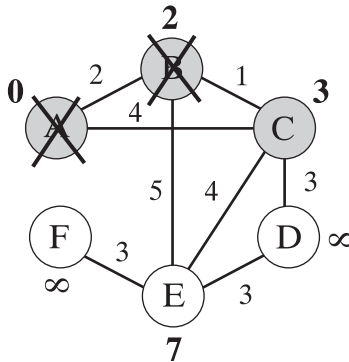


Шаг 1-2. Из непосещённых вершин выберем вершину с минимальной меткой — это вершина B (выделена цветом). Определим метки её соседей (посещённую вершину A не рассматриваем).

$$BC = 2 + 1 = 3; 3 < 4;$$

$$BE = 2 + 5 = 7; 7 < \infty.$$

Вершине C присвоим метку 3, вершине E — метку 7. Вершину B пометим как посещённую.

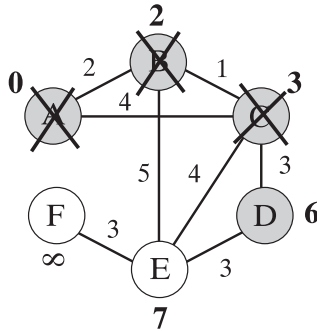


Шаг 1-3. Из непосещённых вершин выберем вершину с минимальной меткой — это вершина C. Определим метки её соседей (посещённые вершины A и B не рассматриваем).

$$CD = 3 + 3 = 6; 6 < \infty;$$

$$CE = 3 + 4 = 7; 7 = 7.$$

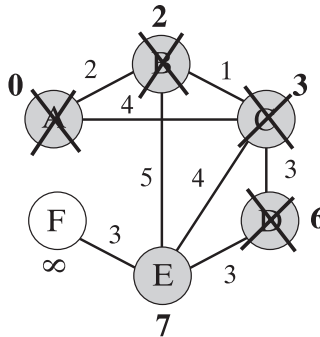
Вершине D присвоим метку 6, метка вершины E не изменится. Вершину C пометим как посещённую.



Шаг 1-4. Из непосещённых вершин выберем вершину D с минимальной меткой. Определим метку её соседа E, посещённые вершины не рассматриваем.

$$AE = 6 + 3 = 9; 9 > 7.$$

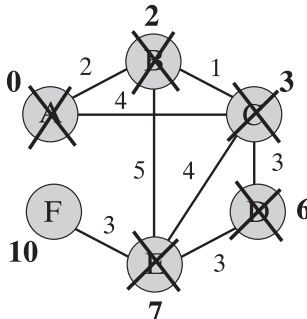
Метка вершины E не изменится. Вершину D пометим как посещённую.



Шаг 1-5. Из непосещённых вершин E и F выберем вершину E с минимальной меткой. Определим метку её соседа F.

$$AF = 7 + 3 = 10; 10 < \infty.$$

Метка вершины F равна 10, это и есть кратчайшее расстояние от вершины A до вершины F. Вершину E пометим как посещённую.



Вершина F не имеет непосещённых соседей, пометим её как посещённую, выполнение алгоритма завершается.

В результате метка каждой вершины соответствует минимальному расстоянию от A до этой вершины.

О т в е т: 2.

См. учебно-справочные материалы: Элементы теории графов, с. 124.

А3 Дан фрагмент таблицы истинности выражения F:

X	Y	Z	F
0	0	1	0
0	1	1	0
1	0	1	1

Каким может быть выражение F?

- 1) $\neg X \vee \neg Y \vee Z$
- 2) $X \wedge \neg Y \wedge Z$
- 3) $X \vee \neg Y \vee Z$
- 4) $\neg X \wedge Y \wedge \neg Z$

Решение. Если бы таблица истинности была задана полностью, т.е. на всех наборах переменных X, Y и Z, можно было бы решить это задание построением КНФ или ДНФ².

Самый простой способ решения этого задания — подстановка значений переменных из строк таблицы истинности в предложенные в ответах выражения и сравнение результата вычисления выражения с соответствующим значением F в заданном фрагменте таблицы истинности. Выражение, значения которого для всех трёх наборов аргументов будут совпадать со значениями, указанными в таблице, и есть правильный ответ.

² Конъюнктивная нормальная форма или дизъюнктивная нормальная форма.

X	Y	Z	F	$\neg X \vee \neg Y \vee Z$	$X \wedge \neg Y \wedge Z$	$X \vee \neg Y \vee Z$	$\neg X \wedge Y \wedge \neg Z$
0	0	1	0	1	0	1	0
0	1	1	0	1	0	1	0
1	0	1	1	1	1	1	0

Значения выражения варианта ответа № 2 совпадают с **F** на всех заданных наборах.

Ответ: 2.

См. учебно-справочные материалы: Нормальные формы логических функций, с. 97.

A4 Для групповых операций с файлами используются **маски имён файлов**. Маска представляет собой последовательность букв, цифр и прочих допустимых в именах файлов символов, в которых также могут встречаться следующие символы:

символ «?» (вопросительный знак) означает ровно один произвольный символ;

символ «*» (звёздочка) означает любую последовательность символов произвольной длины, в том числе символ «*» может задавать и пустую последовательность.

Определите, по какой из масок будет выбрана указанная группа файлов:

```
сеееес.с
ebcb.res
ecbe.res
еееесс.exe
```

- 1) *с*.??**
- 2) ?с*. *?

- 3) *с*. *
- 4) *сс*. *

Решение. Определим закономерность в именах файлов этой группы. Во-первых, обратим внимание на расширения этих файлов. Один из них (сееееес.с) имеет расширение, со-

стоящее только из одного символа. Таким образом, ответ 1 неверный, так как он подразумевает наличие двух и более символов в расширении. Рассмотрим ответ 2. Он подразумевает, что на втором месте в имени файла стоит символ «с». Но это неверно для двух файлов: `ebcb.res` и `eeeecc.exe`. Рассмотрим ответ 4. Он тоже неверен в силу того, что только в двух файлах из четырёх встречается двойной символ «с». Таким образом, верным является ответ 3: символ «*» допускает любое количество символов как перед, так и после символа «с» в имени файла, а также в расширении.

Можно составить таблицу, в которой запишем, удовлетворяет ли имя файла маске. Если хотя бы одно имя файла не соответствует маске, остальные имена для этой маски проверять не надо.

	1) *c*.??**	2) ?c*.*?	3) *c*.*	4) *cc*.*
<code>cccccc.c</code>	-	+	+	+
<code>ebcb.res</code>		-	+	-
<code>ecbe.res</code>			+	
<code>eeeecc.exe</code>			+	

Ответ: 3.

См. учебно-справочные материалы: Поиск файлов, с. 36.

A5 Автомат получает на вход два трёхзначных числа. По этим числам строится новое число по следующим правилам:

1. Записывается результат сложения значений старших разрядов двух заданных чисел.
2. К нему дописывается результат сложения значений средних разрядов этих чисел по такому правилу: если он меньше первой суммы, то второе полученное число приписывается к первому слева, иначе — справа.
3. Итоговое число получают приписыванием справа к полученному после второго шага числу суммы значений младших разрядов исходных чисел.

Пример. Исходные трёхзначные числа: 138, 212. Поразрядные суммы: 3, 4, 10. Результат: 3410.

Определите, какое из предложенных чисел может быть результатом работы автомата.

- 1) 3507 2) 13412 3) 91216 4) 13197

Решение. Решим задачу с помощью рассуждений. Итоговое число состоит из трёх частей, каждая из которых представляет собой сумму двух однозначных чисел.

- Сумма цифр старших разрядов принадлежит отрезку $[2, 18]$. Минимальные цифры в старших разрядах трёхзначного числа равны 1, минимальная сумма старших разрядов равна 2. Максимальные цифры в разрядах равны 9, максимальная сумма равна 18.
- Суммы цифр остальных разрядов принадлежат отрезку $[0, 18]$.

По заданию можно определить, что вторая часть числа должна быть больше первой. Рассмотрим каждое из предложенных чисел:

- 1) 3507 — варианты представления числа: 35 0 7, 3 50 7 и 3 5 07. В первых двух вариантах представления значения 35 и 50 больше 18. В третьем варианте десятичная запись числа начинается с 0 (07). Ответ неверный.
- 2) 13412 — разобьём число на части, учитывая, что каждая часть числа должна быть меньше 19: 13, 4, 12. $13 > 4$, ответ неверный.
- 3) 91216 — можно разбить число на 9, 12 и 16. $9 < 12$. Следовательно, вариант ответа может быть верным.
- 4) 13197 — каким бы образом мы не разбили это число на три части, одна из них всегда будет больше 18, значит, вариант ответа неверный.

Ответ: 3.

См. учебно-справочные материалы: Основы алгоритмизации, с. 161.

А6 База данных гостиницы состоит из трёх связанных таблиц. Ниже приведены фрагменты этих таблиц.

Таблица «Бронирование гостиничных номеров»

ID_брони	ID_постояльца	ID_номера	Дата заезда	Дата выезда	Количество суток
11117	D01	131	24.01.11	16.02.11	24

Продолжение таблицы

ID_брони	ID_постояльца	ID_номера	Дата заезда	Дата выезда	Количество суток
11125	D13	114	19.03.11	30.03.11	11
11141	D34	131	02.06.11	30.06.11	28
11152	D05	111	12.04.11	24.04.11	12
11158	D34	211	09.07.11	05.09.11	58
11169	D45	476	17.04.11	25.04.11	9
11174	D67	211	28.09.11	18.10.11	20
11178	D13	425	15.11.11	30.11.11	15

Таблица «Постояльцы»

ID_постояльца	Фамилия	Пол	Номер паспорта
D01	Васильев	М	1280567
D05	Смирнова	Ж	3560246
D13	Иванова	Ж	1730567
D34	Фёдоров	М	1956230
D45	Ильин	М	1854634
D67	Андреева	Ж	9672549

Таблица «Гостиничные номера»

ID_номера	Телефон	Количество комнат	Категория	Стоимость в сутки
111	7575-111	2	люкс	2500,00
114	7575-114	1		1700,00
131	7575-131	2	люкс	2700,00
211	7575-211	1		2000,00
425	7575-425	3	люкс	4000,00
476	7575-476	1		1500,00

По данным в представленных фрагментах таблиц определите, какую выручку получит гостиница за номера категории «люкс», заказанные клиентами женского пола.

- 1) 30 000 2) 60 000 3) 90 000 4) 148 700

Решение. Выясним по таблице «Гостиничные номера» **ID_номера** категории «люкс».

ID_номера	Телефон	Количество комнат	Категория	Стоимость в сутки
111	7575-111	2	люкс	2500,00
131	7575-131	2	люкс	2700,00
425	7575-425	3	люкс	4000,00

В таблице «Постояльцы» находим **ID_постояльца** женского пола:

ID_постояльца	Фамилия	Пол	Номер паспорта
D05	Смирнова	ж	3560246
D13	Иванова	ж	1730567
D67	Андреева	ж	9672549

В таблице «Бронирование гостиничных номеров» находим строки, в которых **ID_номера** и **ID_постояльца** равны полученным.

ID_брони	ID_постояльца	ID_номера	Дата заезда	Дата выезда	Количество суток
11152	D05	111	12.04.11	24.04.11	12
11178	D13	425	15.11.11	30.11.11	15

Считаем выручку, которую получит гостиница:

$$12 \cdot 2500 + 15 \cdot 4000 = 90\ 000.$$

Ответ: 3.

См. учебно-справочные материалы: Реляционная база данных, с. 244.

A7 В динамической (электронной) таблице приведены значения количества студентов, изучающих иностранные языки на разных курсах, и количество отличников из них. Укажите изучаемый иностранный язык, для которого общий процент отличников максимальный.

Курс	Количество студентов	Язык			
		английский	французский	немецкий	китайский
1	всего	160	140	130	50
	из них отличников	25	15	30	3
2	всего	120	120	120	30
	из них отличников	24	12	12	7
3	всего	130	110	150	20
	из них отличников	12	13	20	5
4	всего	100	70	125	10
	из них отличников	10	15	17	4
Итого	всего	510	440	525	110
	из них отличников	71	55	79	19

1) английский

3) немецкий

2) французский

4) китайский

Решение. Следует оценить отношение отличников к общему количеству студентов, изучающих соответствующий язык.

Получим:

английский: $71 / 510 \approx 0,14$;

французский: $55 / 440 \approx 0,125$;

немецкий: $79 / 525 \approx 0,15$;

китайский: $19 / 110 \approx 0,17$.

Частное от деления довольно сложно вычислить без калькулятора. Предлагаем оценить отношение следующим образом:

- 1) английский: $71 \cdot 7 = 497$; $71 \cdot 8 = 568$; следовательно, $71 \cdot 7 < 510 < 71 \cdot 8$, и $1 / 8 < 71 / 510 < 1 / 7$;
 2) французский: $55 / 440 = 1 / 8$;
 3) немецкий: $1 / 7 < 79 / 525 < 1 / 6$;
 4) китайский: $1 / 6 < 19 / 110 < 1 / 5$.

Наиболее успешно студенты изучают китайский язык.

Ответ: 4.

A8 Производится одноканальная монозвукозапись с частотой дискретизации 16 кГц и 24-битным разрешением. Запись длится 1 минуту, её результаты записываются в файл, сжатие данных не производится. Какое из приведённых ниже чисел наиболее близко к размеру полученного файла, выраженному в мегабайтах?

- 1) 0,2 2) 2 3) 3 4) 4

Решение. Измерение звука производится 16 000 раз в секунду или $16000 \cdot 60$ раз в минуту. Для сохранения результата каждого измерения требуется 24 бит = 3 байт. Всего $16\,000 \cdot 60 \cdot 3$ байт. Вычислим размер файла в мегабайтах:

$$16\,000 \cdot 60 \cdot 3 / 2^{20} = 2^7 \cdot 125 \cdot 2^2 \cdot 15 \cdot 3 / 2^{20} = 125 \cdot 15 \cdot 3 / 2^{11} \approx \approx 6000 / 2048 \approx 3.$$

Ответ: 3.

См. учебно-справочные материалы: Кодирование звуковой информации, с. 70.

A9 Для кодирования некоторой последовательности, состоящей из букв **А**, **Б**, **В**, **Г** и **Д**, решили использовать неравномерный двоичный код, позволяющий однозначно декодировать двоичную последовательность, появляющуюся на приёмной стороне канала связи. Использовали код: **А** – 1, **Б** – 000, **В** – 001, **Г** – 011. Укажите, каким кодовым словом должна быть закодирована буква **Д**.

Длина этого кодового слова должна быть наименьшей из всех возможных. Код должен удовлетворять свойству однозначного декодирования.

- 1) 00 2) 0101 3) 11 4) 010

Решение. В задании используется префиксный код — код со словами переменной длины, в котором ни одно кодовое слово не является началом другого кодового слова. Он позволя-

ет однозначно декодировать сообщение без специального разделения кодов символов.

Проверим, являются ли предложенные ответы началом заданных кодовых слов или, наоборот, заданные коды — началом кодов, предложенных в ответах.

- 1) 00 — является началом кода символа Б (000).
- 2) 0101 — не является началом ни одного из заданных кодов символов, ни один из заданных кодов не является началом этого кода.
- 3) 11 — код символа А (1) является началом этого кода.
- 4) 010 — не является началом ни одного из заданных кодов символов, ни один из заданных кодов не является началом этого кода.

Таким образом, кодом буквы Д может быть 0101 и 010. По условию задания требуется выбрать кодовое слово минимальной длины, следовательно, верным ответом является код 010.

О т в е т : 4.

См. учебно-справочные материалы: Префиксные коды, с. 8.

A10 Какое из приведённых имён удовлетворяет логическому условию:

- (первая буква гласная) →
 → (последняя буква согласная ∧ в слове больше 5 букв)?
- 1) ПОЛИНА 2) ГЛЕБ 3) СЕРГЕЙ 4) НИКИТА

Решение. Введём обозначения высказываний:

- А — первая буква гласная,
 В — последняя буква согласная,
 С — в слове больше 5 букв.

Логическое условие можно записать в виде выражения
 $\neg A \rightarrow (B \wedge C) = A \vee B \wedge C$.

Проще всего решить данную задачу с помощью построения таблицы истинности:

	А	В	С	$B \wedge C$	$A \vee B \wedge C$
ПОЛИНА	0	0	1	0	0
ГЛЕБ	0	1	0	0	0
СЕРГЕЙ	0	1	1	1	1
НИКИТА	0	0	1	0	0

Правильный ответ – СЕРГЕЙ.

Ответ: 3.

См. учебно-справочные материалы: Основные законы алгебры логики, с. 92.

A11 При сдаче экзамена в университете в память компьютера заносится индивидуальный буквенно-цифровой код студента и его оценка. Код состоит из 12 символов. В качестве символов используют 26 букв английского алфавита и десятичные цифры в любом порядке. При этом используют посимвольное кодирование и все символы кодируются одинаковым и минимально возможным количеством битов. Оценка — число от 0 до 10 — также кодируется с использованием минимально возможного и одинакового количества битов. Код студента вместе с его оценкой записывается минимально возможным и целым числом байтов. Определите объём файла, содержащего такую информацию, после сдачи экзамена 48 студентами.

- 1) 624 байт 2) 480 байт 3) 384 байт 4) 528 байт

Решение. Количество возможных символов в индивидуальном коде студента $26 \text{ букв} + 10 \text{ цифр} = 36$. Определим минимально необходимое количество битов для кодирования 36 различных символов. При мощности алфавита, равной двум (для кодирования числа используются цифры 0 и 1), количество разрядов (бит) для кодирования одного из 36 различных символов определяется по формуле:

$$\lceil I \rceil = \lceil \log_2 N \rceil$$

где I — количество битов для кодирования одного символа, а N — количество возможных различных символов. Запись $\lceil X \rceil$ означает, что X округляется до ближайшего большего целого. Для кодирования одного символа нам потребуется $\lceil \log_2 36 \rceil = 6$ бит. На 12 символов кода необходимо $6 \cdot 12 = 72$ бит. Для кодирования оценки необходимо $\lceil \log_2 11 \rceil = 4$ бит. Итого на каждого студента $72 + 4 = 76$ бит, или 10 байт. Информация о 48 студентах займёт $48 \cdot 10 = 480$ байт.

Ответ: 2.

См. учебно-справочные материалы: Измерение информации, с. 11.

A12 В программе используется одномерный целочисленный массив **A** с индексами от 0 до 10 и целочисленные переменные **tmp** и **i**. Ниже представлен фрагмент программы, записанный на разных языках программирования, в котором значения элементов сначала задаются, а затем меняются.

Бейсик	Паскаль
<pre>FOR i=0 TO 10 A(i)=i NEXT i FOR i=0 TO 9 STEP 3 TMP = A(i) A(i) = A(i+1) A(i+1) = TMP NEXT i</pre>	<pre>for i:=0 to 10 do A[i]:=i; for i:=0 to 9 do if (i mod 3)=0 then begin tmp := A[i]; A[i]:=A[i+1]; A[i+1]:=tmp; end;</pre>
Си	Алгоритмический язык
<pre>for (i=0;i<=10;i++) A[i]=i; for (i=0;i<10;i+=3) { tmp= A[i]; A[i]=A[i+1]; A[i+1]=tmp; }</pre>	<pre>нц для i от 0 до 10 A[i]:=i кц нц для i от 0 до 9 шаг 3 tmp := A[i] A[i]:=A[i+1] A[i+1]:=tmp кц</pre>

Чему будут равны элементы этого массива после выполнения фрагмента программы?

- 1) 3 1 2 6 4 5 9 7 8 0 10 3) 1 0 2 4 3 5 7 6 8 10 9
 2) 1 0 3 2 5 4 7 6 9 8 10 4) 0 1 2 3 4 5 6 7 8 9 10

Решение. В задаче используются два оператора цикла с параметром. В первом цикле

Бейсик	Паскаль
<pre>FOR i=0 TO 10 A(i)=i NEXT i</pre>	<pre>for i:=0 to 10 do A[i]:=i;</pre>

Си	Алгоритмический язык
<pre>for (i=0;i<=10;i++) A[i]=i;</pre>	<pre>нц для i от 0 до 10 A[i]:=i кц</pre>

массив А заполняется числами от 0 до 10:

i	0	1	2	3	4	5	6	7	8	9	10
A[i]	0	1	2	3	4	5	6	7	8	9	10

Во втором цикле

Бейсик	Паскаль
<pre>FOR i=0 TO 9 STEP 3 TMP = A(i) A(i) = A(i+1) A(i+1) = TMP NEXT i</pre>	<pre>for i:=0 to 9 do if (i mod 3)=0 then begin tmp := A[i]; A[i]:=A[i+1]; A[i+1]:=tmp; end;</pre>
Си	Алгоритмический язык
<pre>for (i=0;i<10;i+=3) { tmp= A[i]; A[i]=A[i+1]; A[i+1]=tmp; }</pre>	<pre>нц для i от 0 до 9 шаг 3 tmp := A[i] A[i]:=A[i+1] A[i+1]:=tmp кц</pre>

параметр цикла i изменяется с шагом 3 и принимает значения 0, 3, 6, 9. В теле цикла меняются местами значения переменных $A[i]$ и $A[i+1]$, т. е. значения $A[0]$ и $A[1]$, $A[3]$ и $A[4]$, $A[6]$ и $A[7]$, $A[9]$ и $A[10]$:

	A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]	A[10]
исх	0	1	2	3	4	5	6	7	8	9	10
i=0	1	0	2	3	4	5	6	7	8	9	10
i=3	1	0	2	4	3	5	6	7	8	9	10
i=6	1	0	2	4	3	5	7	6	8	9	10
i=9	1	0	2	4	3	5	7	6	8	10	9

Значения $A[2] = 2$, $A[5] = 5$, $A[8] = 8$ остались без изменений.
В результате получим:

i	0	1	2	3	4	5	6	7	8	9	10
A[i]	1	0	2	4	3	5	7	6	8	10	9

Ответ: 3.

См. учебно-справочные материалы: Одномерные массивы, с. 218.

A13 Система команд исполнителя РОБОТ, «живущего» в прямоугольном лабиринте на клетчатой плоскости:

вверх	вниз	влево	вправо
-------	------	-------	--------

При выполнении любой из этих команд РОБОТ перемещается на одну клетку соответственно: вверх \uparrow , вниз \downarrow , влево \leftarrow , вправо \rightarrow .

Четыре команды проверяют истинность условия отсутствия стены у каждой стороны той клетки, где находится РОБОТ:

сверху свободно	снизу свободно	слева свободно	справа свободно
-----------------	----------------	----------------	-----------------

Цикл ПОКА $\langle \text{условие} \rangle$ команда выполняется, пока условие истинно, иначе происходит переход на следующую строку.

Если РОБОТ начнёт движение в сторону стены, то он разрушится и программа прервётся.

Сколько клеток лабиринта соответствует требованию, что, выполнив предложенную программу, РОБОТ уцелеет и остановится в той же клетке, с которой он начал движение?

НАЧАЛО

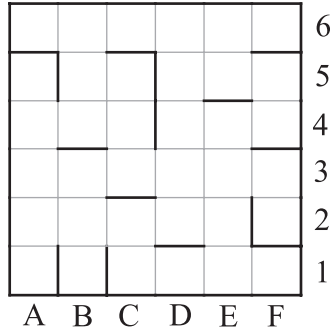
ПОКА <слева свободно> вверх

ПОКА <сверху свободно> вправо

ПОКА <справа свободно> вниз

ПОКА <снизу свободно> влево

КОНЕЦ



1) 0

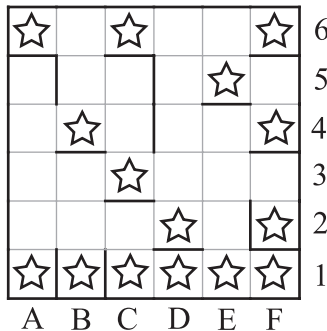
2) 1

3) 2

4) 4

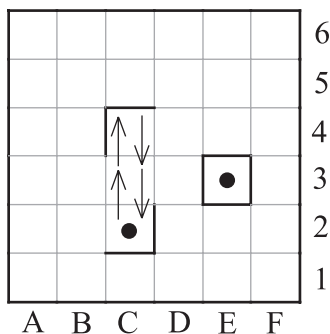
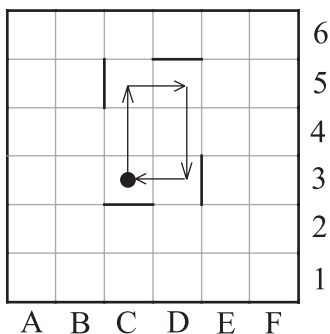
Решение. Очевидным, но трудоёмким способом решения данной задачи является проверка всех клеток поля. Сократим перебор.

Заметим, что РОБОТ заканчивает выполнение алгоритма в клетке, ограниченной стеной снизу, следовательно, надо проверить только клетки, имеющие стенку снизу. Отметим эти клетки:

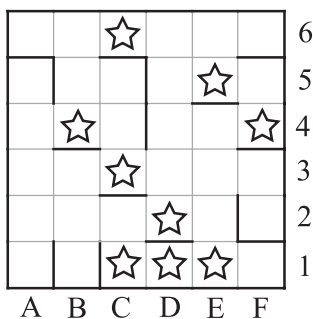


На рисунке отмечено максимальное количество клеток, которые потребуется проверять. Продолжим рассуждения, чтобы сократить число проверяемых клеток.

Для того чтобы РОБОТ успешно вернулся в начальную клетку, требуется выполнение алгоритма, задающего траекторию движения РОБОТА. В общем случае траекторией является прямоугольник, движение начинается вверх и идёт по часовой стрелке. Возможные виды траекторий показаны на рисунках ниже.

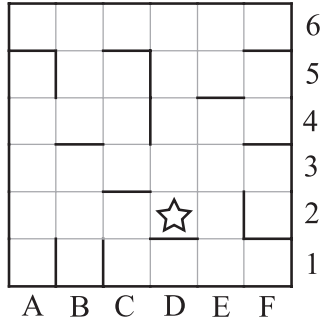


Отметим, что при выполнении заданного алгоритма РОБОТ всегда разрушается, если клетка ограничена тремя стенами. Удалим отметки с таких клеток:



Для оставшихся отмеченных клеток проверим выполнение алгоритма:

B4	Вверх B5 — вправо C5 — влево B5 — разрушился при движении влево
C1	Вправо D1 — разрушился при движении вниз
C3	Вверх C4, C5 — разрушился при движении вверх
C6	Разрушился при движении вверх
D1	Разрушился при движении вверх
D2	Вверх D3, D4 — вправо E4 — вниз E3, E2 — влево D2
E1	Вверх E2, E3, E4, разрушился при движении вверх
E5	Вверх E6, разрушился при движении вверх
F4	Вверх F5, разрушился при движении вверх



Отмеченной осталась одна клетка.

Ответ: 2.

См. учебно-справочные материалы: Формальный исполнитель РОБОТ, с. 179.

A14 Определите, какое число будет напечатано в результате работы следующей программы (для вашего удобства программа представлена на четырёх языках):

Бейсик	Паскаль
<pre> Module A14 Sub Main() Dim d, a, b, t, M, R As Double a = -2 : b = 3 d = 0.1 t = a: M = a: R = F(a) While t <= b If F(t) < R Then M = t R = F(t) End If t = t + d End While Console.Write(M) End Sub Function F(ByVal x As Double) As Double Return (x-1) * (x-3) End Function End Module </pre>	<pre> Program A14; Uses crt; Var d,a,b,t,M,R :real; Function F(x:real):real; begin F:=(x-1)*(x-3); end; BEGIN a:=-2; b:=3; d:=0.1; t:=a; M:=a; R:=F(a); while t<=b do begin if (F(t)<R) then begin M:=t; R:=F(t); end; t:=t+d; end; write(M); END. </pre>

Си	Алгоритмический язык
<pre>#include <stdio.h> double F(double x) { return (x-1)*(x-3); } void main() { double d, a, b, t, M, R; a = -2; b = 3; d = 0.1; t = a; M = a; R = F(a); while (t<=b) { if (F(t)<R) { M = t; R = F(t); } t = t + d; } printf("%.0f", M); }</pre>	<pre>алг A14 нач вещ d, a, b, t, M, R a:= -2; b:= 3 d:= 0.1 t:=a; M:=a; R:=F(a) нц пока t<=b если F(t)< R то M:= t; R:= F(t) все t:= t + d кц ВЫВОД M кон алг вещ F(вещ x) нач знач := (x-1)*(x-3) кон</pre>

1) -1

2) -2

3) 2

4) 3

Решение. Для решения этого задания нерационально использовать трассировку программы, так как тело цикла выполняется $(b - a) / d + 1 = 51$ раз.

Следует понять, что именно вычисляется в программе. Это поможет определить результат её работы.

Переменные a , b , d используются для организации цикла.

В программе используется функция пользователя $F(x)$, которая возвращает значение выражения $(x-1)*(x-3)$. Значение аргумента в теле функции не изменяется³.

В цикле вычисляется минимальное значение функции $F(t)$ на интервале $[a, b]$ с шагом d . В переменной R сохраняется значение функции $F(t)$, претендующее на минимальное. В переменной M сохраняется значение аргумента t , при котором функция принимает минимальное значение.

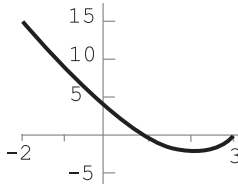
Таким образом, в программе определяется значение аргумента функции F , при котором функция принимает минимальное значение, на отрезке $[-2;3]$.

³ Параметры функции передаются в неё по значению.

Запишем функцию:

$$F(x) = (x - 1)(x - 3) = x^2 - 4x + 3.$$

Получили уравнение параболы ($px^2 + qx + k$), ветви которой направлены вверх ($p > 0$), а вершина находится в точке $x_{\text{в}} = -q / 2p = 4 / 2 = 2$, $x_{\text{в}} \in [-2; 3]$.



В вершине параболы её значение минимально. Таким образом, на печать будет выведено число 2.

О т в е т : 3.

См. учебно-справочные материалы: Подпрограммы, определяемые разработчиком, с. 232.

Часть 2

Ответом к заданиям части 2 (В1–В14) является число, последовательность букв или цифр. Впишите ответы сначала в текст работы, а затем перенесите их в бланк ответов № 1 справа от номера соответствующего задания, начиная с первой клеточки, без пробелов, запятых и других дополнительных символов. Каждую букву или цифру пишите в отдельной клеточке в соответствии с приведёнными в бланке образцами.

В1 Автоматическое устройство осуществило перекодировку информационного сообщения на русском языке длиной 75 символов, первоначально записанного в 16-битной кодировке Unicode, в 8-битный код КОИ-8. На сколько битов уменьшилась длина сообщения? *В ответе запишите только число.*

Решение. Сообщение состоит из 75 символов. Первоначально оно имело объём $75 \cdot 2$ байт, после перекодирования — $75 \cdot 1$ байт. Следовательно, объём сообщения уменьшился на

$$75 \cdot 2 - 75 = 75 \text{ байт} = 600 \text{ бит.}$$

О т в е т: 600.

См. учебно-справочные материалы: Кодирование текстовой информации, с. 60.

В2 У исполнителя Вычислитель две команды, которым присвоены номера:

1. приписать 3 справа
2. вычесть 1

Первая из них приписывает к числу на экране справа цифру 3, вторая вычитает из него 1.

Запишите порядок команд в алгоритме получения из числа 7 числа 532, содержащем не более 5 команд, указывая только номера команд (например, 21222 — это алгоритм:

2. вычесть 1
1. приписать 3 справа
2. вычесть 1
2. вычесть 1
2. вычесть 1

который преобразует число 4 в число 30).

Если таких алгоритмов более одного, запишите любой из них.

Решение. Построим дерево выполнения команд. В результате выполнения команды 1 получим $10 \cdot x + 3$, где x — число, к которому применяется команда.

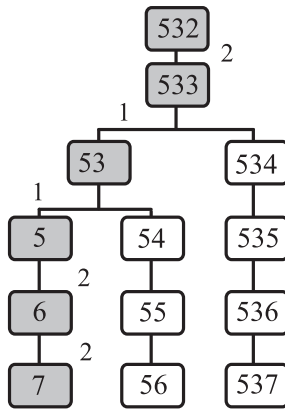
Команду 1 можно применить ко всем числам. Но обратную к ней команду «вычесть 3 и разделить на 10» можно применить только к числам, в которых последняя цифра 3. Таких чисел в десять раз меньше, и обратная команда может применяться в 10 раз реже. Команда 2 и обратная ей могут применяться ко всем числам. Вывод: количество ветвей дерева, построенного от результата 532 к исходному числу 7 с использованием команд, обратных заданным, меньше, чем при построении дерева от заданного числа 7 к результату 532.

Обратные команды:

1. вычесть 3 и разделить на 10
2. прибавить 1

Из каждого узла дерева могут выходить две ветви: левая соответствует выполнению обратной команды 1, правая —

выполнению обратной команды 2. Если команда 1 не может быть выполнена, из узла выходит одна ветвь, соответствующая команде 2.



Запишем последовательность команд получения из числа 7 числа 532 (от листа к корню дерева).

Ответ: 22112.

См. учебно-справочные материалы: Формальный исполнитель Вычислитель, с. 185.

В3 Определите, что будет напечатано в результате работы следующего фрагмента программы:

Бейсик	Паскаль
<pre>Dim a, b a = 75 b = 10 - a Mod 2 While a - 4 * b > 2 If a > 30 Then a = a - 30 Else : a = a + b + 10 End If End While Console.Write(a)</pre>	<pre>var a,b: integer; BEGIN a:=75; b:=10 - a mod 2; while a-4*b>2 do begin if a>30 then a:=a-30 else a:=a+b+10; end; write(a); END.</pre>

Си	Алгоритмический язык
<pre> { int a,b; a = 75; b = 10 - a % 2; while (a-4*b)>2 if(a>30) a = a - 30; else a = a + b + 10; printf("%d", a); } </pre>	<pre> нач цел a,b a:=75 b:=10-mod(a,2) нц пока a-4*b>2 если a>30 то a:=a-30 иначе a:=a+b+10 все кц ВЫВОД a кон </pre>

Решение. В алгоритме используется циклическая структура с предусловием, в теле цикла выполняется структура ветвления — полная форма команды «если». Составим трассировочную таблицу первого вида:

№	Команда или условие (логическое выражение)	Вычисление правой части команды присваивания или условия (логического выражения)	a	b
1	a := 75	75	75	
2	b := 10 - mod(a, 2)	10 - 1 = 9		9
3	a - 4*b > 2	75 - 4*9 = 75 - 36=39; (39 > 2) = да		
4	a > 30	(75 > 30) = да		
5	a := a - 30	75 - 30 = 45	45	
6	a - 4*b > 2	45 - 4*9 = 9; (9 > 2) = да		
7	a > 30	(45>30) = да		
8	a := a - 30	45 - 30 = 15	15	
9	a - 4*b > 2	15 - 4*9 = 15 - 36 = -21; (-21 > 2) = нет		
Результат:			15	9