

**Указания для экспертов по проверке заданий
с развернутым ответом по информатике и ИКТ**

1. Общие рекомендации

При проверке правильности решений следует учитывать, что образцы решений, приведенные в указаниях по проверке отдельных задач, не являются единственно возможными. Таким образом, то, что решение ученика не совпадает с решением в образце, само по себе не означает, что решение экзаменуемого неверное. При оценке следует руководствоваться указаниями по оцениванию, примеры решений являются лишь подспорьем для проверяющего.

В указаниях по оцениванию для каждой задачи подробно описаны критерии, при которых следует за решение ставить тот или иной балл. Ниже даны рекомендации по проверке отдельных задач и рассмотрены некоторые сложные случаи.

2. Рекомендации по отдельным задачам

Задача 24

При проверке задачи 24 следует последовательно проверить, насколько выполнены следующие **четыре** действия:

- 1) указано, что выведет программа при конкретной входной последовательности;
- 2) указан пример последовательности, при которой программа работает правильно;
- 3) исправлена первая ошибка;
- 4) исправлена вторая ошибка.

ВНИМАНИЕ! Экзаменуемому достаточно указать один пример таких входных данных. Указывать полное описание всех таких данных не требуется. Описания, приведенные в критериях по оцениванию (см. «Замечания для проверяющего»), предназначены только для проверяющего. Если при выполнении второго задания экзаменуемый указал несколько вариантов входных данных, то задание считается выполненным, если для **ВСЕХ** указанных входных данных программа дает нужный результат.

При проверке каждого из заданий на исправление ошибок (третье и четвертое действия) следует убедиться, что:

- а) каждое внесенное исправление затрагивает только одну строку в программе;

- б) указан такой новый вариант строки, что при исправлении другой ошибки получается правильная программа.

Каждое из этих действий оценивается отдельно. Обратите внимание: исправления, указанные экзаменуемым, могут отличаться от приведенных в критериях, но тем не менее быть верными.

Задача 25

1. Определите, на каком языке записан алгоритм в проверяемой работе, и при необходимости наведите справки о синтаксисе избранного экзаменуемым языка программирования.
2. Сравните описание алгоритма с имеющимися образцами и в случае совпадения оцените его в соответствии с рекомендациями.
3. Если описание алгоритма не совпадает с образцами, а ошибки в описании алгоритма с первого взгляда не видны, осуществите формальное исполнение алгоритма с тестовыми примерами исходных данных. Длину тестового массива следует сократить до четырех–шести элементов. При составлении тестов необходимо особенно тщательно проверять «критические» случаи, например когда элементы массива одинаковы или изначально упорядочены. Оцените правильность полученных результатов.
4. В случае алгоритма, представленного на русском языке или в виде блок-схемы, необходимо оценить возможность выполнения этого алгоритма человеком и уровень детализации алгоритма (должны выполняться требования дискретности, детерминированности и результативности).
5. Обратите внимание на правильность инициализации переменных. Учтите, что приведенное в решении значение может не быть единственно возможным.
6. Обязательно проверьте, присутствует ли вывод ответа.
7. Отдельно проверьте, обрабатывается ли случай, что искомого числа в данном массиве может и не быть (если иное не сказано в условии задачи).
8. При оценке алгоритма отметьте все ошибки, упомянутые в критериях оценивания. В случае, если таких ошибок две или более, сразу снижайте оценку до 0 баллов.
9. Не допускайте произвольного ужесточения критериев оценивания. Не вводите дополнительных ограничений. Не оценивайте синтаксические ошибки, «стиль» программирования, аккуратность записи, наличие комментариев, отступов и прочие важные, но не проверяемые данной задачей вещи.
10. Не забывайте, что эффективность алгоритмов в данной задаче не оценивается, поэтому не следует снижать оценку за решение, в котором, например, для поиска максимума используется сортировка.

Задача 26

1. При описании стратегии выигрывающего игрока должно быть указано, каким будет ход этого игрока при любой игре противника. Это может быть сделано по-разному: в виде словесного описания (как это сделано в решении из критериев оценивания для заданий 1 и 2), в виде таблицы ходов или дерева всех возможных партий (см. решение для задания 3). Наличие слов «указываем ход при любом ответе противника» или тому подобных не обязательно. Важно, чтобы из решения точно следовало, как именно должен ходить выигрывающий игрок в любой возможной ситуации.
2. Обозначения на рисунке могут отличаться от использованных в примере решения, например не обязательно использовать пунктирные линии.

Задача 27

I. Особенности задачи 27.

1. В задаче есть два задания (А и Б). Соответственно, ученик может представить две программы. В каждой из программ должно быть указано, решением какого из заданий она является.

Если в работе представлена одна программа, то в ней также должно быть указано, решением какого из заданий она является.

Если в работе представлено больше двух программ, то проверяются две последние программы.

Если ученик не указал, к какому заданию относится программа, или можно предположить, что ученик ошибся в идентификации программ, следуйте приведенным ниже инструкциям.

2. Ученик представил только одну программу. Программа рассматривается как решение задания Б и оценивается по соответствующим критериям.

3. Ученик представил две программы, но указание задания есть только для одной из программ. Вторая программа рассматривается, как ответ на оставшееся задание.

4. Ученик представил две программы, ни для одной из них задание не указано. Первая (по порядку в представленных учеником материалах) рассматривается, как ответ на задание А, а вторая - как ответ на задание Б.

5. **НАПОМИНАЕМ:** *Итоговый балл за задачу - это **больший** из баллов, полученных учеником за каждое из заданий А и Б*

II. Проверка отдельной программы.

Рекомендуем выполнять проверку программы в следующей последовательности.

1. Определите язык программирования, на котором написана программа. Программы, написанные на языках программирования, отличных от Паскаля, Алгоритмического языка, Бейсика и Си, тоже должны быть оценены. При необходимости эксперт может воспользоваться справочной литературой, а также обратиться к консультанту или председателю предметной комиссии. Подсчитайте количество таких синтаксических ошибок в программе, которые не мешают понять ее логику. Систематически встречающаяся ошибка считается за одну. Так, например, если вместо круглых скобок ученик в записи условий использовал везде квадратные, то это считается за одну ошибку.
2. Рассмотрите реализацию алгоритма и определите, верна ли она в целом. Если верна, определите, есть ли в ней легко диагностируемые и исправляемые алгоритмические ошибки (см. критерии для выставления 3 баллов при выполнении задания Б). Определите количество таких алгоритмических ошибок (если они есть).
3. Если ошибок мало и программа оценивается 3–4 баллами, то оцените эффективность предложенного решения. Определите, сколько времени (шагов алгоритма) и сколько памяти требуется программе экзаменуемого.
4. Далее руководствуйтесь указаниями по оцениванию.

3. Разбор некоторых сложных для оценивания ситуаций

Задача 24

№	Типичная проблемная ситуация	Решение
1	Участником экзамена указаны в качестве ошибочных верные строки, содержащие, по его мнению, синтаксические ошибки или неверный тип переменных.	Констатируется, что верная строка указана учащимся как ошибочная, и этот факт учитывается при оценивании по критериям.
2	При выполнении первого и/или второго элемента задания экзаменуемый привел несколько числовых ответов, часть из которых верна, а часть нет.	Соответствующий пункт задания признается выполненным неверно.
3	В исправленной строке допущены синтаксические ошибки, не искажающие замысел автора.	Соответствующий пункт задания признается выполненным верно.
4	Строка с ошибкой не выписана, как требуется в задании, а указана иным образом, например, номером.	Если способ, которым учащийся указал строку, позволяет её однозначно идентифицировать, приравниваем эту ситуацию к выписыванию строки.
5	Учащийся заново переписал текст программы или её фрагменты из нескольких строк, внося исправления.	Если исправления касались только строк с ошибками, засчитываем этот элемент как выполненный. В противном случае, считаем, что элемент не выполнен.

Задача 25

Основная сложность при оценивании задачи 25 состоит в том, что в условии не установлено четко, каким образом алгоритм обработки массива может быть записан. Разрешение использовать естественный язык автоматически приводит к употреблению неточных и расплывчатых формулировок, оборотов «аналогично ищем максимальный элемент», «производим подсчет этих чисел, выводим результат» и т.д. При оценке алгоритмов, записанных на естественном языке, основным критерием должна быть возможность их формального исполнения, т.е. описание должно быть максимально приближено к записи команд, соответствующих основным операторам языков программирования: должно быть указано, в какой последовательности выполняются инструкции, каково условие завершения цикла и т.д. При этом, например, фраза «Для каждого элемента массива мы выполняем сравнение его с переменной...» может рассматриваться как вполне приемлемое описание цикла с параметром.

При рассмотрении записей алгоритмов на естественном языке следует очень внимательно проследить, как экзаменуемый описывает операцию присваивания. Наряду с формулой «Присваиваем переменной m значение очередного элемента массива» может употребляться и конструкция «Присваиваем значение очередного элемента массива переменной m » (часто употребляется конструкция «записываем в ...»). Не стоит снижать оценку за некоторую нечеткость терминологии, если из описания алгоритма видно, что экзаменуемый ясно представляет себе его выполнение. Если же алгоритм не может быть формально исполнен без каких-либо особых разъяснений (формально можно описать это как вызов процедуры или вспомогательного алгоритма), то такой алгоритм должен быть оценен 0 баллов. Пример такого описания: «Сначала находим максимальное значение элементов массива, а затем...». Иногда экзаменуемые пытаются отделаться только общими указаниями, что может быть оценено только в 0 баллов. Например, фраза «Находим максимальные элементы массива и подсчитываем их число» должна быть оценена в 0 баллов как недостаточная для формального исполнения.

№	Типичная проблемная ситуация	Решение
1	Отсутствует в явном виде инициализация счетчика или сумматора найденных элементов (пар, троек и т.д) массива. При этом могут быть комментарии, указывающие на возможность автоматической инициализации счетчика нулевым значением.	В соответствии с критериями оценивания отсутствие инициализации в тексте программы считается ошибкой.
2	Вместо того, чтобы написать фрагмент программы соответствующий многоточию	В случае верной программы оценка не снижается. При наличии алгоритмических

	в условии, учащийся пишет программу целиком.	ошибок (в том числе допущенных при переписывании части программы из условия) оценка снижается в соответствии с критериями оценивания.
3	Формат вывода результата несколько отличается от предписанного в задании, например, добавляется слово «ответ:» или при нулевом количестве искомых элементов печатается «нет» вместо 0.	Оценка не снижается

Задача 26

При описании выигрышной стратегии для определенной позиции указан один из возможных выигрывающих ходов, но не указаны другие возможные выигрывающие ходы. Это не является ошибкой.

№	Типичная проблемная ситуация	Решение
1	При описании стратегии учащийся пропускает простые арифметические выкладки, например, вместо «Ваня удвоит количество камней во второй куче, при этом он получит в обеих кучах $10+25 \times 2 > 55$ камней и выиграет» пишет «Ваня удвоит количество камней во второй куче и выиграет».	Оценка не снижается.
2	Учащийся описывает верную стратегию, но обосновывает её утверждениями, содержащими логические ошибки, например, «При любом ходе Пети, Ване следует удвоить количество камней во второй куче, потому что, игрок, который удваивает, всегда выигрывает».	Ответ не засчитывается как верный.

Задача 27

1. В решении может присутствовать опечатка, в результате которой оно становится синтаксически или логически неверным ($a[i]$ вместо $a[j]$, $m = a[i]$ вместо $m := a[i]$ и т.п.). Если из анализа решения видно, что это именно опечатка (например, из комментариев или если в большинстве аналогичных случаев автор программы не ошибся), то ее НЕ следует рассматривать как синтаксическую или логическую ошибку.
2. Технические программистские ошибки (например, отведение недостаточного количества памяти, необнуление значения переменной или выход за границу массива, некорректное использование стандартных функций соответствующего языка программирования) следует приравнивать к синтаксическим, если они не порождены ошибками в алгоритме решения.

3. Программа может быть написана на языке программирования, отличном от тех, которые используются и приводимых в КИМах примерах программ, например на языках C++, C#, Java, Perl, PHP и т.д. При этом допускается использование библиотек, описанных в стандартах соответствующего языка. Например, при использовании языка C++ допускается использование средств стандартных библиотек (библиотеки STL). Могут использоваться, например, версии языков Паскаль и Бейсик, отличные от использованных в примерах решений. Во всех таких случаях экзаменуемый обязан явно указать, какой именно версией какого языка он пользуется. Если например, в решении не указана версия языка, а использованное экзаменуемым средство доступно не во всех версиях языка, считается, что экзаменуемый допустил ошибку. Чтобы разобраться в подобных ситуациях эксперт может воспользоваться справочной литературой, а также обратиться к консультанту или председателю предметной комиссии.
4. В тексте задания есть фраза «Перед текстом программы кратко опишите используемый вами алгоритм решения задачи». Отсутствие такого описания (при наличии правильной программы) не является основанием для снижения оценки, задание НЕ контролирует умение экзаменуемого составлять описания. Назначение описания – облегчить работу проверяющего эксперта. Согласно указаниям по оцениванию эксперт должен решать, являются ли ошибки, допущенные экзаменуемым, случайными опечатками или они показывают некомпетентность экзаменуемого. Описание может помочь проверяющему понять текст программы, зная, что именно собирается в ней делать экзаменуемый. Хорошее описание может быть основанием для выставления 1 балла в случае плохой реализации. В критериях оценивания на 1 балл указано: «Программа, возможно, неверно работает при некоторых входных данных, но по приведенному тексту решения ясно, что экзаменуемый понимает, из каких этапов должно состоять решение задачи». Здесь «текст решения» включает в себя и описание, о котором идет речь.
5. В задаче 27 использование учащимся в решении подпрограмм (функций) из стандартных библиотек вполне допустимо при условии, что они правильно описаны (если это требуется правилами языка) и правильно вызываются, т.е. нет ошибок в количестве, порядке и типе аргументов. См. также п. 3

№	Типичная проблемная ситуация	Решение
1	Учащийся допускает в программе алгоритмическую ошибку, которая не входит в приведенный в критериях оценивания перечень допустимых ошибок на два или три балла.	В соответствии с критериями оценивания программа оценивается не более, чем в 1 балл, независимо от её возможных остальных достоинств.

Заполнение протокола проверки

Номера столбцов в протоколе проверки экспертом заданий с развернутым ответом											
1	2	3	4	5	6	7	8	9	10	11	12
24	25	26	27								
Номера заданий в варианте											

Внимание! При выставлении баллов за выполнение задания в «Протокол проверки ответов на задания бланка № 2» следует иметь в виду, что **если ответ отсутствует** (нет никаких записей, свидетельствующих о том, что экзаменуемый приступал к выполнению задания), то в протокол проставляется «**X**», а не «0».