

Подготовка учащихся к ГИА по Информатике по теме «Алгоритмизации и программирование»

Основной государственной экзамен представляет собой форму государственной итоговой аттестации, проводимой в целях определения соответствия результатов освоения обучающимися основных образовательных программ основного общего образования требованиям федерального государственного образовательного стандарта. Для указанных целей используются контрольные измерительные материалы, представляющие собой комплексы заданий стандартизированной формы.

Экзаменационная работа охватывает основное содержание курса информатики в соответствии с ФГОС. Охвачен наиболее значимый материал.

Содержательная линия «Алгоритмизация и программирование» представлена в ОГЭ тремя заданиями:

- Задание №5. Анализ алгоритмов для Калькулятора.
- Задание №6. Анализ программ с ветвлениями
- Задание №15_1. Составление программы для исполнителя Робот или задание №15_2. Составление программы на обработку потока данных

Основная цель заданий, представленных в ОГЭ по содержательной линии «Алгоритмизация и программирование» проверить:

- На уровне воспроизведения знаний: понятие алгоритма, его свойства, способы записи, основные алгоритмические конструкции.
- На уровне сформированности умений применять свои знания в стандартной ситуации, это: использовать стандартные алгоритмические конструкции для построения алгоритмов для формальных исполнителей и формально исполнять алгоритмы, записанные на естественном и алгоритмическом языках.
- На уровне сформированности умений применять свои знания в новой ситуации: разрабатывать алгоритм для формального исполнителя или на языке программирования с использованием условных инструкций и циклов, а также логических связей при задании условий.

Задание №5. Анализ алгоритмов для Калькулятора. Ученики должны уметь анализировать простые алгоритмы для конкретного исполнителя с фиксированным набором команд. Уровень сложности данного задания – базовый.

Пример задания.

У исполнителя Бета две команды, которым присвоены номера:

1. прибавь b
2. умножь на 3

(b – неизвестное натуральное число) Выполняя первую из них, Бета увеличивает число на экране на b , а выполняя вторую, умножает это число на 3. Программа для исполнителя Бета – это последовательность номеров команд. Известно, что программа 21212 переводит число 6 в число 282. Определите значение b .

Для решения этой задачи необходимо последовательно выполнить заданную последовательность команд к числу 6, составив математическое выражение, из которого потом и найти b .

$$\begin{aligned}((6 * 3 + b) * 3 + b) * 3 &= 282 \\ ((18 + b) * 3 + b) * 3 &= 282 \\ (54 + 3b + b) * 3 &= 282 \\ (54 + 4b) &= 94 \\ 4b &= 94 - 54 \\ b &= 10\end{aligned}$$

Как и в большинстве простых заданий, основные ошибки происходят из-за торопливости и невнимательности. Чтобы уменьшить вероятность ошибки, обязательно учите детей после решения делать проверку.

Задание №6. Анализ программ с ветвлениями. Формально исполнять алгоритмы, записанные на языке программирования. Уровень сложности данного задания – базовый.

Пример задания.

Дана программа:

Python	Паскаль
<pre>x = int(input()) y = int(input()) if y <= 100 or x > 90: print("ДА") else: print("НЕТ")</pre>	<pre>var x, y: integer; begin readln(x); readln(y); if (y <= 100) or (x > 90) then writeln('ДА') else writeln('НЕТ') end.</pre>

Было проведено 10 запусков этой программы, при которых в качестве значений переменных x и y вводились следующие пары чисел:

(105, 100); (95, 110); (100, 95); (95, 90);
(105, 90); (85, 110); (100, 110); (85, 105);
(85, 95); (90, 100)

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

При выполнении заданий, подобных примеру, нет необходимости выполнять программу для каждой представленной в условии пары входных данных. Необходимо провести простой анализ программы, выделив условие печати нужного ответа, и подсчитать, сколько пар удовлетворяют указанному условию. Если же ученики испытывают затруднения при выполнении анализа программы, то конечно же не стоит им запрещать подставлять заданные пары чисел в условный оператор.

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, быть внимательным и правильно выполнить строгое / нестрогое сравнение (если оно есть), а также внимательно смотреть на то какая переменная вводится первой, а какая второй, чтобы не ошибиться при проверке условий.

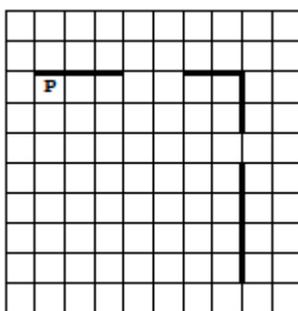
Задание 15 проверяет умение записать формальный алгоритм с использованием инструкций ветвления и цикла. Задание представлено в двух вариантах: для исполнителя «Робот» и для арифметико-логического исполнителя, реализованного в виде системы программирования на языке высокого уровня. Важно отметить, что согласно спецификации экзамена экзаменуемый должен выбрать, какой из вариантов задания он выполняет. Выполнение обоих вариантов не приносит дополнительных баллов экзаменуемому. Данные задания оцениваются экспертами и поэтому учителя должны знать критерии оценивания данных заданий, чтобы работы учеников получили максимальный балл.

Задание 15.1 проверяет умение записать формальный алгоритм с использованием инструкций ветвления и цикла для формального исполнителя «Робот». Алгоритм может быть выполнен в среде формального исполнителя или же записан в текстовом редакторе. Формулировке задания 15.1 предшествует описание команд исполнителя и синтаксиса записи алгоритмических конструкций

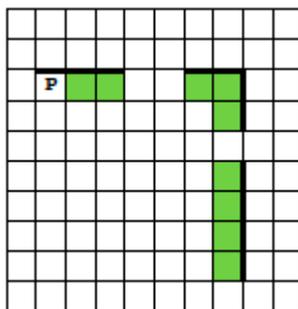
ветвления и цикла, поэтому задание может быть выполнено даже теми школьниками, которые не знакомы с используемым синтаксисом языка, но владеют основами алгоритмизации.

Пример задания.

На бесконечном поле есть горизонтальная и вертикальная стены. Правый конец горизонтальной стены соединён с верхним 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 под 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 ниже горизонтальной стены и левее вертикальной стены, кроме клетки, в которой находится Робот перед выполнением программы. Проходы должны остаться не закрашенными. 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.

Оценивание этого задания производится путём анализа записанного алгоритма, поиска в нём ошибок, проверки, правильно ли алгоритм выполняет задачу и т.д. Прежде всего следует отметить, что записанный алгоритм должен работать при любых длинах горизонтальной и вертикальной стен, а также проходов в них, а не только для того примера, который приведён в условии задачи. Решения, работающие только при каких-то конкретных размерах коридора, оцениваются в 0 баллов, поскольку они решают задачу только для частного случая.

Таким образом, любое правильное решение этой задачи обязательно должно содержать минимум два цикла, перемещающие робота вдоль стен, то есть:

нц пока справа свободно

вправо

кц

и

нц пока не справа свободно

вниз

кц

Первый цикл приведёт Робота в угол двух стен, а второй выведет под стену, находящуюся справа. Однако так как в вертикальной стене есть разрыв, одного этого цикла будет недостаточно. Если такие циклы в алгоритме отсутствуют, то задание решено неверно в любом случае и оценивается в 0 баллов.

Далее необходимо ответить на вопросы:

1. Завершает ли работу проверяемый алгоритм (то есть верно ли, что алгоритм не содержит бесконечных циклов)?
2. Остаётся ли робот цел в результате исполнения алгоритма (то есть верно ли, что робот не разрушается от столкновения со стеной)?
3. Полностью ли робот выполняет поставленную задачу, то есть закрашивает ли все требуемые клетки?

Если ответы на все вопросы утвердительные, то есть алгоритм всегда заканчивает свою работу, робот не разрушается при исполнении алгоритма и полностью выполняет поставленную в условии задачу, то выполнение задания оценивается в 2 балла. При этом не учитываются синтаксические ошибки в записи

алгоритма, то есть проверяется умение составить алгоритм, а не синтаксически правильно его.

Задание оценивается в 1 балл, если закрашены не все клетки, которые должны быть закрашены, или закрашено несколько лишних клеток. Как правило, это одна пропущенная или неправильно записанная команда (например, закрашивается первая клетка, на которой стоит Робот, или закрашивается одна из клеток под проходом сверху или слева от прохода в правой стене, что может быть результатом неправильного чередования команд движения и закрашивания в теле цикла). Выполняя оценивание, данного задания необходимо учитывать, какое количество лишних клеток будет закрашено в том случае, если длины стен и проходов очень большие.

Так, в 0 баллов будет оценено решение, в котором Робот закрашивает все клетки от первоначального положения до стены справа, в том числе все клетки под проходом в горизонтальной стене.

Задание оценивается в 0 баллов, если закрашивается (не закрашивается) более 10 лишних клеток при очень больших размерах стен и проходов или алгоритм изложен неверно. Также в 0 баллов оценивается решение, не имеющее завершения или приводящее к аварийному останову.

Задание 15.2 Задание 15 в варианте 15.2 проверяет умение записать алгоритм на языке программирования. Оно проверяет умения, связанные с созданием простейших программ, содержащих цикл и ветвление внутри цикла, на одном из языков программирования.

Приведём пример задания 15.2.

Напишите программу, которая в последовательности натуральных чисел определяет количество чисел, кратных 4, но не кратных 7. Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число, кратное 4 и не кратное 7. Количество чисел не превышает 1000. Введённые числа не превышают 30 000. Программа должна вывести одно число: количество чисел, кратных 4, но не кратных 7.

Пример работы программы

Входные данные	Выходные данные
4	2

16	
28	
26	
24	

Решением данного задания является программа, записанная на одном из языков программирования. При оценивании необходимо выяснить, правильно ли работает данная программа.

Если программа содержит синтаксические ошибки, и потому её компиляция и запуск невозможны, то задание оценивается в 0 баллов.

Правильная программа должна содержать следующие части:

1. Ввод данных.
2. Нахождение ответа.
3. Вывод ответа.

Программа, которая не содержит ввода данных (например, данные для работы программы задаются константами в её исходном коде) или не содержит вывода ответа, также оценивается в 0 баллов.

Для проверки правильности алгоритма решения задачи, необходимо чтобы программа прошла представленные тесты. Если программа выдаёт правильный ответ, то данный тест считается пройденным, если программа выдаёт неправильный ответ, или не завершает свою работу из-за заикливания, или аварийно завершает свою работу из-за алгоритмической ошибки (например, деления на ноль), то данный тест не засчитывается. Вывод о правильности программы делается в том случае, если программа проходит все тесты.

Оценка в 2 балла за задание 15.2 выставляется, если программа правильная, в том числе выдаёт верный ответ на всех тестах.

Оценка в 1 балл выставляется, если она содержит ошибку, которая приводит к неверному ответу не более чем на одном из тестов.

Оценка в 0 баллов выставляется во всех остальных случаях.

Рекомендации для учеников по выполнению задания 15_1. Обратите внимание, в условии сказано: «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. 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. Для задания 15_2 вышесказанные рекомендации также справедливы.

Тема «Алгоритмизация и программирование» считается одной из самых сложных в школьном курсе информатики, как в теоретическом, так и в практическом плане. Надеюсь, что представленные методические рекомендации позволят учителю качественно подготовить школьников к ОГЭ по информатике.