

Методические рекомендации для учителей, подготовленные на основе анализа типичных ошибок участников ЕГЭ 2016 года по информатике и ИКТ

**Лещинер Вячеслав
Роальдович**

кандидат педагогических наук, ФГБНУ «ФИПИ»,
руководитель федеральной комиссии по разработке
КИМ для ГИА по информатике и ИКТ

**Ройтберг Михаил
Абрамович**

доктор физико-математических наук, член федеральной
комиссии по разработке КИМ для ГИА по информатике и ИКТ,
kim@fipi.ru

Ключевые слова: КИМ ЕГЭ по информатике и ИКТ, основные результаты ЕГЭ по информатике и ИКТ в 2016 г., анализ результатов по блокам содержания, анализ результатов по группам учебной подготовки, статистические характеристики заданий экзаменационной работы.

Контрольные измерительные материалы единого государственного экзамена по информатике и ИКТ позволяют установить уровень освоения выпускниками ФК ГОС среднего (полного) общего образования (приказ Минобрнауки России от 05.03.2004 № 1089). В то же время результаты ЕГЭ признаются образовательными учреждениями высшего профессионального образования как результаты вступительных испытаний по информатике и ИКТ. Именно это двойное назначение экзамена определяет специфику КИМ ЕГЭ как инструмента педагогического измерения.

Содержанием экзаменационной работы охватываются основное содержание курса информатики, важнейшие его темы, наиболее значимый в них материал, однозначно трактуемый в большинстве преподаваемых в школе вариантов курса информатики. Работа содержит как задания базового уровня сложности, проверяющие знания и умения, предусмотренные стандартами базового уровня подготовки по предмету, так и задания повышенного и высокого уровней, проверяющие знания и умения, предусмотренные профильным стандартом. Количество заданий в тесте должно, с одной стороны, обеспечить всестороннюю проверку знаний и умений выпускников, приобретенных за весь период обучения по предмету, и, с другой стороны, обеспечить адекватное ранжирование абитуриентов вузов по уровню подготовки к получению образования по выбранной специальности высшего

профессионального образования. С этой целью в тесте используются задания двух типов: с кратким ответом и с развёрнутым ответом. Задания с кратким ответом в виде числа или строки символов, распознаваемым и проверяемым компьютером, исключают возможность угадывания ответа. Таких заданий в работе 23, они расположены в первой части.

Во второй части работы даётся 4 задания, требующие развёрнутого ответа. Эти, наиболее трудоёмкие и позволяющие экзаменуемым в полной мере проявить свою индивидуальность и приобретённые в процессе обучения умения, задания проверяются экспертами региональных экзаменационных комиссий на основании единых критериев проверки, являющихся частью контрольных измерительных материалов по предмету. Выполнение этих заданий требует значительного времени, и в связи с наличием человеческого фактора при их проверке имеется определённая вероятность ошибки оценивания.

Итак, каждый вариант экзаменационной работы состоит из двух частей и включает в себя 27 заданий, различающихся формой и уровнем сложности.

Часть 1 содержит 23 задания базового, повышенного и высокого уровней сложности. В этой части собраны задания с кратким ответом, подразумевающие самостоятельное формулирование и запись ответа в виде числа или последовательности символов. Задания проверяют материал всех тематических блоков. В первой части 12 заданий относятся к базовому уровню, 10 заданий имеют повышенный уровень сложности, 1 задание — высокий уровень сложности.

Часть 2 содержит 4 задания, первое из которых повышенного уровня сложности, остальные 3 задания высокого уровня сложности. Задания этой части предполагают запись развёрнутого ответа в произвольной форме. Они направлены на проверку сформированности важнейших умений записи и анализа алгоритмов, предусмотренных образовательными стандартами. Последнее задание работы на высоком уровне сложности проверяет умения по теме «Технология программирования».

Задания КИМ оцениваются разным количеством баллов в зависимости от их типа.

Выполнение каждого задания части 1 оценивается в 1 балл. Задание части 1 считается выполненным, если экзаменуемый дал ответ, соответствующий коду верного ответа. За выполнение каждого задания присваивается (в дихотомической системе оценивания) либо 0 баллов («задание не выполнено»), либо 1 балл («задание выполнено»). Ответы на задания части 1 автоматически обрабатываются после сканирования бланков ответов № 1. Максимальное количество первичных баллов, которое можно получить за выполнение заданий части 1, — 23.

Выполнение заданий части 2 оценивается от 0 до 4 баллов. Ответы на задания второй части проверяются и оцениваются экспертами (устанавливается соответствие ответов определённому перечню критериев). Максимальное количество баллов, которое можно получить за выполнение заданий части 2, — 12.

Максимальное количество первичных баллов, которое можно получить за выполнение всех заданий экзаменационной работы, — 35.

Минимальное количество баллов ЕГЭ по информатике и ИКТ, подтверждающее освоение выпускником основных общеобразовательных программ среднего (полного) общего образования в соответствии с требованиями ФК ГОС, составляет 40 баллов по стобальной шкале (установлено Распоряжением Рособнадзора № 3499—10 от 29.08.2012 г.).

В 2015 году КИМ ЕГЭ претерпел значительные изменения по сравнению с КИМ 2014 года. Тогда изменилось деление работы на части, сократилось общее количество заданий и соответственно уменьшилось максимальное количество первичных баллов. Изменился алгоритм перевода первичных баллов в тестовые и уменьшилось до 6 минимальное количество первичных баллов ЕГЭ по информатике и ИКТ, подтверждающее освоение выпускником основных общеобразовательных программ среднего (полного) общего образования.

В 2016 году КИМ ЕГЭ сохранил значительную преемственность с КИМ 2015 года. Основные характеристики работы: количество заданий, сложность заданий, количество первичных баллов и алгоритм перевода первичных баллов в тестовые — остались не-

Таблица 1

Год	Средний тестовый балл	Диапазон тестовых баллов				
		0–20	21–40	41–60	61–80	81–100
2016	56,63	6,89	9	38,11	36,16	9,84
2015	53,99	8,75	11,85	38,57	32,63	8,21
2014	57,79	4,16	8,9	41,93	37,86	7,15

изменными. Отличия коснулись только содержания трёх заданий первой части, в связи с полным отказом от заданий с выбором ответа. Также в связи с этим изменился порядок следования первых 5 заданий 1 части, остальные задания остались на своих местах.

Таким образом, сохранился реализованный в 2015 г. подход укрупнения тематики заданий, сведения близких по тематике и сложности заданий в одну позицию. Такими укрупнёнными были в 2016 г. позиции 4 (хранение информации в компьютере), 6 (формальное исполнение алгоритмов), 7 (технология вычислений и визуализации данных с помощью электронных таблиц) и 9 (скорость передачи звуковых и графических файлов). В КИМ ЕГЭ, использовавшихся на экзамене, в части вариантов были задания по одной из указанных в спецификации тем, в другой части — по смежной теме. Это сильно повысило вариативность вариантов, добавив элемент неопределённости. В спецификации КИМ ЕГЭ 2017 г. укрупнённые позиции сохранены, этот факт необходимо учитывать при подготовке учащихся к экзамену.

Последнее задание 27, проверяющее умение создавать самостоятельные программы, даётся в двух вариантах, более простом, оцениваемом в 2 балла, и более сложном, требующем создания эффективного по памяти и скорости выполнения алгоритма, с оценкой максимально 4 первичных балла.

Спецификация и демонстрационный вариант КИМ ЕГЭ 2017 г. полностью соответствуют КИМ ЕГЭ 2016 г., поэтому результаты ЕГЭ 2016 г. следует обязательно учитывать при подготовке к экзамену будущего года.

Общее количество участников экзамена в 2016 г. — 49 380 чел., число снижается год от года (в 2015 г. — 50 394 чел., в 2014 г. — 53 281 чел., в 2013 г. — 54 897 чел.), но доля

от общего числа участников ЕГЭ более или менее неизменна: чуть выше 7%.

Данные о распределении участников по группам тестовых баллов приведены в табл. 1. Числа, соответствующие диапазонам тестовых баллов, составляют доли в процентах.

Важно отметить, что доля участников, получающих высокие тестовые баллы (81–100) увеличивается год от года (9,84% в 2016 г. против 8,21% в 2015 г. и 7,15% в 2014 г.).

Минимальный балл ЕГЭ 2016 г. составлял 6 первичных баллов, что приравнивалось к 40 тестовым баллам. Доля выпускников, не набравших минимального количества баллов в 2016 г. составила 12,4%, в то время как в 2015 г. она составляла 16,15%. Это, безусловно, позитивный факт, связанный, в первую очередь, с упомянутым выше сокращением доли участников экзамена в регионах со стабильно низкими результатами ЕГЭ по информатике и ИКТ, а также с возросшим в целом уровнем подготовки к экзамену.

Спецификация КИМ ЕГЭ устанавливает три уровня сложности заданий: базовый, повышенный и высокий, при этом для заданий базового уровня примерный интервал выполнения задания предполагается 60–90%; для повышенного уровня результат выполнения должен быть в интервале 40–60%; с заданиями высокого уровня сложности должны справляться менее 40% участников экзамена. Поэтому содержательный анализ результатов ЕГЭ следует начать с определения того, насколько результат выполнения отдельных заданий совпал с их прогнозируемой сложностью.

В работе было 27 заданий, из которых 12 заданий базового уровня сложности, 11 заданий повышенного уровня и 4 задания высоко-

кого уровня сложности. В заданиях базового уровня от экзаменуемого требовалось непосредственно применить конкретные знания, умения и навыки по одной из тем курса. В заданиях повышенного уровня и, в большей степени, высокого уровня необходимо применять знания материала из различных областей и ориентироваться в более или менее новой для выпускника постановке задачи.

Результат выполнения 14 из 23 заданий базового и повышенного уровня (61% заданий) укладывается в границы, установленные для заданий соответствующего уровня сложности (от 60 до 90% выполнения для базового уровня, от 40 до 60% для повышенного уровня). В отличие от 2015 г., в работе не оказалось заданий, которые были бы слишком лёгкими для своего уровня. В 2015 г. таких было 4 задания (2 задания базового уровня и 2 задания повышенного уровня). В работе 2014 г. слишком легких заданий (особенно базового уровня) было ещё больше, так что достигнутый в 2016 г. результат можно считать очень хорошим с точки зрения соответствия номинальной и реальной сложности заданий.

Из 12 заданий базового уровня 7 не вызывают затруднений у большинства сдающих экзамен. Это задания 1–4, 6–8. Они проверяют знания и умения по темам «Двоичная и кратные системы счисления», «Таблицы истинности логических выражений», «Моделирование», «Базы данных и файловые системы», «Формальное исполнение алгоритма», «Электронные таблицы», «Переменные, оператор присваивания, вычислительные алгоритмы». Можно считать, что этот материал, относящийся к базовому содержанию школьного курса информатики, хорошо усвоен выпускниками.

Пять заданий базового уровня сложности вызвали определённые затруднения. Два задания — 10 (53% выполнения) и 9 (46% выполнения) связаны с тематикой равномерного кодирования текстов (10 задание) и звука или растровых изображений (9 задание). И в предыдущие годы задания по этой тематике вызывали затруднения. По сравнению с 2015 годом в выполнении этих заданий наметился некоторый прогресс (в прошлом году процент выполнения задания 10 был 36%, задания 9 — 39%), но пока пока-

затели не дотягивают до соответствующих уровню сложности заданий.

Задание 5 по теме «Неравномерное кодирование текста» в 2016 году было в новом формате, связанном с полным отказом от заданий с выбором ответа. В 2015 году данное задание предполагало выбор одного ответа из четырёх возможных, стояло под номером 1 в варианте и имело процент выполнения 44%. В текущем году задание было в формате, предполагавшем числовой ответ, показатель его выполнения снизился до 41%. Понятие неравномерного кодирования плохо усвоено многими выпускниками, на эту тему, видимо, учителя обращают недостаточно внимания. Стоит отметить, что ошибки при решении этого задания допустили даже некоторые выпускники, набравшие высокий тестовый балл. Задание разобрано далее в разделе 6, учителям следует обратить внимание на эту тему, относящуюся к базовому содержанию информатики.

В методических рекомендациях 2015 г. отмечался крайне низкий результат выполнения задания 11 (25,7% в 2015 г.) по теме «Рекурсивные алгоритмы». В 2016 г. показатель выполнения этого задания возрос до 36%, но этого по-прежнему недостаточно. В разделе 6 методических материалов этого года мы приводим разбор задания ещё раз, так как формирование представления о рекурсивных вызовах процедур и функций относится к числу важных предметных результатов обучения информатике в средней школе, а само по себе понятие рекурсии является фундаментальным.

Ещё одно задание базового уровня, стоявшее в варианте на 12 позиции, проверяет знание выпускниками устройства сети Интернет и адресации в сети. В 2016 году по нему также отмечено снижение показателя выполнения по сравнению с прошлым годом (с 40 до 30%), что, возможно, связано с новизной формулировки задания, использовавшейся в 2016 г. Следует отметить, что содержание не изменилось, изменилась только формулировка. Видимо, данное содержание многими выпускниками было усвоено формально, без глубокого понимания. В разделе 6 даётся разбор задания в формулировке 2016 года. При подготовке выпускников к экзамену учителям следует обратить внимание учащихся, что формули-

ровки задания могут быть различны, но речь идёт об одном и том же явлении, алгоритм построения IP-адреса компьютера по адресу узла в сети и маске сети остаётся неизменным. Это содержание также является базовым, поскольку раскрывает фундаментальные принципы организации всемирной паутины Интернет, которой пользуются все современные люди, независимо от профессии и образования.

Таким образом, подводя итоги выполнения экзаменационной работы в части базового содержания курса, следует отметить, что примерно 40% этого содержания усвоено участниками экзамена недостаточно хорошо. Учителям при подготовке к экзамену прежде всего следует обратить внимание на отработку содержания, проверяемого первыми 12 заданиями КИМ. Нет смысла работать на повышенном уровне, если базовое содержание не усвоено. Также можно ещё раз отметить, что значительная часть экзаменуемых при подготовке ориентируется на тренировку решения конкретного типа заданий, приведённого в демоверсии КИМ ЕГЭ, а не на полноценное усвоение изучаемого материала. Это год от года приводит к снижению результатов выполнения отдельных заданий в том случае, если они даются в новых, непривычных формулировках.

Из 11 заданий повышенного уровня сложности 6 были выполнены с соответствующим уровнем результатом (процент выполнения в диапазоне от 40 до 60%). 4 задания, хотя и были выполнены с недостаточным результатом, но показатель их выполнения повысился по сравнению с 2015 годом. Среди них задание 16 по теме «Системы счисления» и задание 18 по теме «Преобразование логических выражений». Только одно задание 13 показало незначительное снижение показателя выполнения с 42,8% в 2015 г. до 39,4% в 2016 г. Это может быть связано со слегка изменённой формулировкой по сравнению с прошлым годом.

В работе присутствуют 4 задания высокого уровня сложности, одно из которых (23) предполагает краткий ответ в виде числа и оценивается в дихотомической системе, а три других — задания 25–27 с развёрнутым ответом, предполагающие оценивание из нескольких баллов. Среди всех заданий высокого уровня сложности наименьший процент

выполнения (7,3%) имеет задание 23. Этот результат соответствует ожиданиям разработчиков, так как задание 23 рассматривается ими как технически наиболее сложное задание варианта, рассчитанное на наиболее подготовленных выпускников, претендующих на оценку в 100 баллов. Другое, традиционно сложное для выполнения задание 27, проверяющее умение написать самостоятельную программу на языке программирования, в 2016 году, как и в 2015-м, давалось в двух вариантах. Задание А с максимальной оценкой 2 балла представляет собой достаточно формальное техническое упражнение по программированию, и, в принципе, может быть выполнено любым выпускником профильного класса, изучавшим программирование. Задание Б является обобщением задания А на ситуацию с большим количеством исходных данных и требует написания эффективной по времени и памяти программы. С нашей точки зрения число учащихся, получивших за задание 27 оценку 2 балла (3,7%) явно недостаточно и свидетельствует о том, что в массе своей выпускники средней школы не в состоянии написать компьютерную программу, реализующую описанный на естественном языке в условии задания алгоритм вычисления. Ещё 5,7% участников экзамена получили за это задание либо 3, либо 4 балла, то есть справились в той или иной степени с заданием Б. Почти 7% участников получили за задание 27 один балл, показав понимание подходов к написанию программы, но не сумев её завершить. Таким образом, какие-то баллы за задание 27 получили 16,2% экзаменуемых.

С политомическими заданиями 25 и 26 в полном объёме (на высший для задания балл) справились соответственно 33% и 32% участников экзамена. Это хороший результат для задания высокого уровня сложности.

Таким образом, результат выполнения экзаменационной работы в целом в 2016 г. несколько повысился по сравнению с 2015 г., но при этом сохранилось большинство особенностей, недостатков и проблем в подготовке выпускников, отмечавшихся в предыдущие годы.

Так, задание 18 повышенного уровня выполнено в среднем с показателем 20,2%, что почти в 2 раза выше прошлогодних 11,3%, но всё равно недостаточно (при разбросе

по вариантам от 5 до 19%). Задание, проверяющее это содержание (преобразование импликации в логических выражениях), существует в ЕГЭ с 2005 г., но начиная с 2012 г. оно планомерно усложняется. До 2014 г. задание было в группе заданий с выбором ответа, его перенос в прошлом году в категорию заданий с кратким ответом в сочетании с вводом дополнительного содержания в ряде вариантов (функция поразрядной конъюнкции двоичных чисел) привёл к существенному снижению результатов. Задание было проанализировано в методических рекомендациях 2015 г. и в 2016 г. дано в более простой формулировке. Тем не менее, задание остаётся одним из самых сложных в варианте. В рекомендациях 2016 г. мы повторяем разбор задания в формулировке 2015 г. и дополнительно приводим разбор одного из заданий текущего года.

В задании 14 в 2015 г. был описан новый исполнитель Редактор, что, возможно, явилось причиной недостаточно высоких результатов в выполнении этого задания в 2015 году. В 2016 году задание использовало тот же, уже знакомый исполнитель, поэтому 53% его выполнения в текущем году следует признать хорошим показателем для задания повышенного уровня. При подготовке обучающихся к экзамену 2017 года следует учитывать, что, согласно спецификации, это задание предполагает применение знаний в новой для выпускника ситуации (практически это означает, что в задании может быть описан неизвестный экзаменуемым исполнитель, поэтому проверяется умение разобраться в системе команд исполнителя и исполнить алгоритм для него). Задание 16 было похоже на задания 2014 и 2015 гг., было проанализировано в методических рекомендациях. В этом году результаты были выше, чем в предыдущие годы (33,4%), но пока они не дотягивают до необходимого уровня.

Последнее в варианте задание высокоуровня сложности 27, проверяющее умение написать самостоятельную программу на языке программирования, в 2016 г., как и в 2015 г., давалось в двух вариантах. Задание А с максимальной оценкой 2 балла представляет собой достаточно формальное техническое упражнение по программированию, и, в принципе, может быть вы-

полнено любым выпускником профильного класса, изучавшим программирование. Задание Б является обобщением задания А на ситуацию с большим количеством исходных данных и требует написания эффективной по времени и памяти программы. С нашей точки зрения, число учеников, получивших за задание 27 оценку 2 балла (3,7%), явно недостаточно, и свидетельствует о том, что в массе своей выпускники средней школы не в состоянии написать компьютерную программу, реализуя описанный на естественном языке в условии задания алгоритм вычисления. Ещё 5,7% участников экзамена получили за это задание либо 3, либо 4 балла, то есть справились в той или иной степени с заданием Б. Почти 7% участников за задание 27 получили 1 балл, показав понимание подходов к написанию программы, но не сумев её завершить. Таким образом, какие-то баллы за задание 27 получили 16,2% экзаменуемых, что недостаточно.

Итоги ЕГЭ 2016 г. ещё раз показали, что основной причиной провалов по отдельным заданиям является перекося подготовки в сторону механистического решения известных моделей заданий в ущерб изучению фундаментального содержания. В случае отдельных тем, возможно, затруднения экзаменуемых связаны с недостаточным вниманием, уделённым этим темам в процессе преподавания. В частности, вызывает тревогу низкий показатель выполнения задания 27 на самостоятельное программирование.

Вся выборка участников ЕГЭ статистически по результатам экзамена была разделена на 4 группы: не набравших минимального балла, набравших 40–60 тестовых баллов, набравших 61–80 тестовых баллов и набравших высокое (от 81 до 100) число тестовых баллов. Рис. 1 иллюстрирует доли указанных групп в общей совокупности выпускников.

Группу 1 в основном составляют участники, не имеющие даже минимальной подготовки по предмету. В группе 2, наиболее многочисленной, находятся учащиеся, работающие на базовом уровне. Группа 3 — выпускники, достигшие повышенного уровня подготовки. И, наконец, немногочисленную группу 4 составляют экзаменуемые, показавшие высокие результаты. По сравнению с 2015 г. доли групп 2 и 4 возросли, а групп 1 и 3 снизились.

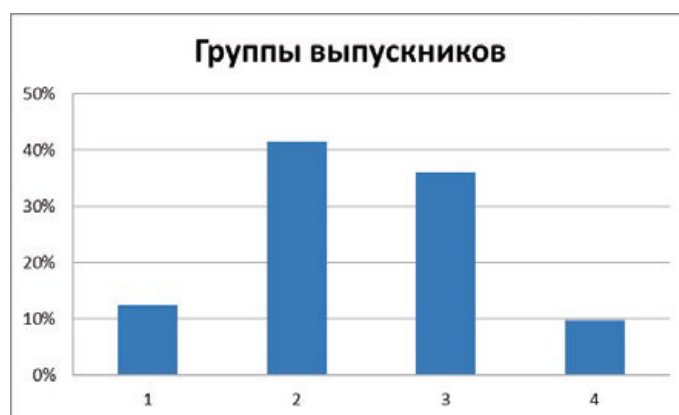


Рис. 1. Распределение участников по уровням подготовки

Группу 1 в основном составляют участники, пришедшие на экзамен случайно, не готовившиеся к экзамену целенаправленно и плохо представляющие себе содержание экзаменационных заданий. Сравнительно лучше других эти участники выполняют задания 4 и 6, которые можно выполнить, опираясь на здравый смысл и следуя инструкции, содержащейся в формулировке задания. Задания простые, но требующие знания терминов и понятий информатики, такие как задания 1 и 7, выполняются этими участниками экзамена неудовлетворительно. С заданием 8, требующим умения прочесть и формально исполнить базовые операторы языка программирования, справились только 30% учащихся из этой группы. Из заданий с развёрнутым ответом только задание 26 выполняется 7% участников группы с минимальным результатом в 1 балл (из 3 возможных).

Выявить обучающихся с подобным уровнем подготовки в условиях школьного класса достаточно просто, это делается обычной контрольной работой. Школьники этой группы вряд ли смогут записать натуральное число в двоичной системе (и наоборот, прочесть двоичную запись числа), затруднятся исполнить алгоритм, записанный на алгоритмическом языке или языке программирования, содержащий оператор присваивания и конструкцию ветвления. Также они не смогут различить относительные и абсолютные ссылки в электронных таблицах. Любое задание на воспроизведение школьных знаний будет выполнено учащимися этой группы, скорее всего, неудовлетворительно.

Для перехода из группы 1 в группу 2 этим учащимся требуется элементарная подготовка к экзамену, хотя бы знакомство с демо-версией КИМ ЕГЭ. Даже небольшой опыт решения заданий базового уровня, основанного на изучении соответствующего материала и освоении проверяемых умений, приведёт к удовлетворительному результату и позволит преодолеть минимальную границу баллов ЕГЭ.

Группа 2 оказалась в этом году самой многочисленной. Её участники работают на базовом уровне, но, к сожалению, выполняют не все задания базового уровня сложности с требуемым результатом. 7 заданий (1–4, 6–8) выполняются ими уверенно, остальные вызывают затруднения. Среди заданий повышенного уровня чуть лучше, но всё равно недостаточно (процент выполнения около 40%) выполняются задания 14, 15 и 17, которые можно решить как специфическими методами информатики, так и опираясь на здравый смысл и общеучебные умения. Задание 26 на то или иное количество баллов (от 1 до 3) выполняют более половины, хотя максимальный балл за него получили только 11%. Чётко видны пробелы в подготовке — неравномерное кодирование (задание 5 — показатель выполнения 28%), рекурсия (задание 11 — 15%), измерение количества информации (задания 9 и 10 — показатели 27% и 36% соответственно). Задание 12 выполнили 14% участников этой группы.

Из четырёх заданий с развёрнутым ответом каких-то значимых результатов участники этой группы достигают при выполнении

только двух заданий: 24 и 26, но и здесь результат намного ниже требуемого.

Выделение этой группы учащихся в составе класса осуществляется посредством диагностических работ в формате ЕГЭ. Выше указаны аспекты содержания, по которым у учащихся данной группы имеются пробелы: «Неравномерное кодирование», «Кодирование растровых изображений», «Адресация в сети Интернет», «Обработка массивов», «Функции в алгоритмах». Учащиеся этой группы скорее всего затруднятся выполнить задание 20, так как не «узнают» реализованный там алгоритм. Они не очень уверенно будут оперировать с натуральными числами, записанными в позиционных системах счисления, отличных от десятичной, могут допускать ошибки при решении комбинаторных задач. В целом усвоение и воспроизведение знаний и умений этой группой учащихся осуществляется формально, без опоры на понимание фундаментальных основ предмета.

Важнейшей задачей учителя на этапе формирующего обучения является профилактика появления этой группы обучающихся. В отличие от участников экзамена из первой группы, которые просто не имеют интереса к предмету и в силу этого «пропускают материал мимо ушей», обучающиеся второй группы вполне мотивированы на изучение информатики. Учитель должен своевременно выявлять возможные пробелы в подготовке, не допуская их формирования. Важно добиться глубокого понимания со стороны учащихся каждой темы, каждого раздела учебного предмета, дать школьникам достаточную практику применения полученных знаний и освоенных умений при решении заданий разных типов и моделей. Полезно на этом этапе использовать тематические сборники заданий в формате ЕГЭ, не отказываясь от решения заданий в формате экзаменов прежних лет, при том что заданий подобных моделей в экзамене 2017 года точно не будет. Так, в открытом банке заданий ЕГЭ содержится огромный корпус заданий с выбором ответа, решение которых будет безусловно эффективно при изучении отдельных тем курса.

Если принадлежность учащегося ко второй группе выявилась в результате диагностической работы на этапе итогового повто-

рения в 11-м классе, оптимальным будет совместное составление учителем и учеником индивидуального плана ликвидации пробелов в подготовке. При этом задача учителя — добиться полного овладения учащимся содержания каждого узкого аспекта подготовки, научить учащихся решать не только конкретное задание, аналогичное имеющемуся в демоверсии, но весь корпус возможных заданий, проверяющих данное содержание. Важнее, чтобы выпускник пришёл на экзамен, твёрдо зная, к решению каких заданий по спецификации он подготовлен полностью и какой первичный балл он должен получить в любом случае, независимо от того, насколько знакомыми ему будут те или иные формулировки заданий.

Конечно, педагогическое мастерство учителя в решении данной задачи играет не меньшую роль, чем уровень мотивации ученика. Именно учитель должен хорошо представлять себе, какие аспекты подготовки проверяет то или иное задание КИМ ЕГЭ, видеть весь корпус возможных типов и моделей заданий, отделять важное содержание от факультативного, показать учащемуся подходы, которые используют авторы КИМ ЕГЭ при конструировании заданий.

Численность данной группы в 2016 г. по сравнению с 2015 г. возросла, группа стала наиболее многочисленной. Произошло это не только за счёт сокращения группы 1, но, к сожалению, за счёт перехода части выпускников из группы 3 в группу 2. Желательно, чтобы на экзамене 2017 г. значительная часть выпускников из этой группы перешла в группу 3, что произойдёт, если они будут выполнять почти все задания базового уровня сложности. Вместе с тем следует признать, что полного исчезновения группы 2 не произойдёт никогда, тем более, что это и не требуется. Среди сдающих ЕГЭ по информатике и ИКТ есть прослойка учеников, изучающих информатику на базовом уровне и планирующих поступление в высшие учебные заведения с невысоким проходным баллом. Уровень подготовки, соответствующий второй группе, показывает, что экзаменуемые могут продолжить обучение в области, связанной с информатикой, при условии, что это обучение начнётся с повторения основ. Поэтому они заслуженно получают по итогам ЕГЭ положительную оценку.

Группа 3 в 2016 г. уменьшилась по сравнению с 2015 г. и утратила статус наиболее многочисленной. Это участники, готовившиеся к экзамену, изучавшие информатику на профильном или углублённом уровне, имеющие опыт решения задач в формате ЕГЭ. От группы 2 их отличает уверенное выполнение заданий 9, 10, 12, 13, 14, 15, 17, 19, 20, 21 и 22 первой части и заданий 24, 25 и 26 с развёрнутым ответом. Вызывают затруднения уже упоминавшиеся задания базового уровня 5 и 11, а также задание повышенного уровня 16 (запись чисел в позиционных системах счисления) — с каждым из них справилось от 52 до 56% членов группы. Результат выполнения задания 12 в этой группе составляет 44%. Задания 18, 23 и 27 выполняются неудовлетворительно (процент выполнения ниже 30).

Отличие этих учеников от учеников наиболее сильной, четвёртой, группы — в уровне способности справляться с новыми формулировками, именно решать задачи, а не выбирать решение среди заранее известных. Диагностика этих умений — важная задача ЕГЭ, который, среди прочего, должен быть инструментом по отбору абитуриентов в ведущие вузы страны.

В то же время способность к творческому мышлению, свободному владению материалом тоже может быть предметом обучения. Эффективный способ повышения индивидуальных результатов для учащихся этой группы — тренировка по решению заданий с нестандартными формулировками, заданий, требующих применения знаний в новой ситуации. Кроме того, следует ликвидировать лакуны в подготовке по отдельным темам (рекурсия, поразрядная конъюнкция двоичных чисел, преобразование логических выражений).

Основной резерв повышения результатов экзамена у этой группы выпускников находится на этапе итогового предэкзаменационного повторения в 11-м классе, так как именно хорошая учебная работа на этапе формирующего обучения обеспечила их попадание в данную успешную группу. Учителю следует по результатам диагностики в формате ЕГЭ определить для каждого учащегося из группы имеющиеся пробелы в подготовке, а затем составить индивидуальные комплекты тренировочных зада-

ний для их ликвидации. Комплекты должны обеспечить формирование опыта применения знаний и умений в новой, нестандартной ситуации.

Учащимся из третьей группы надо чётко оценивать свои силы и не ставить невыполнимых задач. Например, многие участники группы 4 испытывают затруднения при выполнении задания 27. Приобретение опыта самостоятельного программирования требует значительного времени, ставить эту задачу, не перейдя из третьей группы в четвёртую, преждевременно. Также нет смысла учиться выполнять задание 23, если выпускник испытывает затруднения при выполнении задания 18.

Наконец, **группа 4** — это наиболее подготовленные участники экзамена, самая малочисленная группа (менее 10% участников). Эта группа увеличилась по сравнению с экзаменом 2015 г. вдвое, что, безусловно, следует расценивать как позитивное изменение. Выпускниками из этой группы освоены проверяемое содержание и основные группы умений. Определённые затруднения у них вызывают лишь задания 23 и 27 — наиболее сложные в экзамене. Надо отметить, что задание 18, вызвавшее затруднение в 2015 г., в текущем году было выполнено участниками из этой группы с хорошим показателем в 77%. Основным резервом повышения индивидуальных результатов является тренировка по созданию оригинальных программ для решения практических задач — умение, проверяемое последним и наиболее трудоёмким 27-м заданием, правильное выполнение которого приносит выпускнику 4 первичных балла.

Подобная тренировка должна осуществляться опытным учителем, так как самостоятельно проверить качество выполнения программы и её эффективность учащийся обычно не в состоянии. Также важной является последовательность предъявления заданий на программирование, которая должна обеспечить формирование устойчивого умения писать правильные и эффективные программы для решения разнообразных задач тех типов, которые проверяются на ЕГЭ. Неоценимую помощь учителю в подборе таких заданий может оказать открытый банк заданий ЕГЭ прошлых лет.

Анализ результатов ЕГЭ по информатике из года в год показывает, что появление новой формулировки задания вызывает довольно резкое снижение результатов по сравнению с предыдущим годом. Однако уже в следующем году результаты идут вверх, и через пару лет, когда к формулировке все привыкают, оказываются на первоначальном уровне. С учётом того, что объективная сложность заданий не изменяется и основные характеристики участников ЕГЭ по информатике и ИКТ также остаются неизменными, логично предположить, что основной причиной ухудшения результатов по отдельным заданиям являются недостатки в подготовке выпускников. Наша гипотеза состоит в том, что многие учителя при подготовке школьников к ЕГЭ сосредотачиваются на тренировке учащихся в решении заданий, аналогичных заданиям, опубликованным в демонстрационном варианте КИМ, в ущерб фундаментальному изучению предмета. Шок от необычной формулировки задания, получаемый экзаменуемыми, приводит к потере баллов и недостаточно высоким результатам.

Кроме того, анализ выполнения отдельных заданий КИМ ЕГЭ по информатике и ИКТ показывает, что ряд тем, относящихся к базовому содержанию курса, оказывается усвоен на ненадлежащем уровне. На эти темы и их содержание учителям следует обратить особое внимание. Разберём 5 заданий базового уровня сложности, выполненных основной массой участников экзамена с недостаточно высоким результатом.

Задание 10 продолжает линию заданий ЕГЭ, проверяющих знание учащимися комбинаторной формулы зависимости количества возможных кодовых слов от мощности алфавита и длины слова. Это фундаментальное базовое содержание, составляющее теоретическую основу курса и, безусловно, присутствующее во всех учебных программах и учебниках. Напомним здесь основные положения алфавитного подхода к измерению количества информации.

1) Пусть A — упорядоченное множество из N элементов, тогда для кодирования каждого элемента двоичным кодом, например, путём нумерации в двоичной системе счисления, требуется $\log_2 N$ двоичных разрядов

(бит). Объём информации I , содержащейся в сообщении о том, что выбран какой-либо элемент этого множества, равен, соответственно, $\log_2 N$ бит. Если N не является целой степенью 2, то число $\log_2 N$ не является целым, и $I = [\log_2 N + 1]$, то есть происходит округление в большую сторону. При решении задач, если $\log_2 N$ не является целым числом, I можно найти как $\log_2 N'$, где N' ближайшая к N степень двойки, такая что $N' > N$.

2) (Следует из предыдущего.) Если некоторый алфавит содержит M символов, то информационный объём одного символа этого алфавита в сообщении равен $\log_2 M$. Для того чтобы найти информационный объём сообщения, состоящего из символов этого алфавита, следует $\log_2 M$ умножить на количество символов в сообщении.

3) С помощью n двоичных разрядов (бит) можно закодировать двоичным кодом все элементы множества мощностью 2^n (то есть состоящего из 2^n элементов). Информационный объём одного символа, обозначающего элемент данного множества, будет равен n .

Пример задания 10

Игорь составляет таблицу кодовых слов для передачи сообщений, каждому сообщению соответствует своё кодовое слово. В качестве кодовых слов Игорь использует 5-буквенные слова, в которых есть только буквы П, И, Р, причём буква П появляется ровно 1 раз. Каждая из других допустимых букв может встречаться в кодовом слове любое количество раз или не встречаться совсем. Сколько различных кодовых слов может использовать Игорь?

Решение: Буква П появляется в слове 1 раз, поэтому надо определить, сколько всего существует слов, составленных из двух символов, длиной 4 буквы, и полученное число умножить на 5 (так как букву П можно добавить к каждому из этих слов в 5 различных местах). Из двух символов можно составить $2^4 = 16$ различных четырёхбуквенных слов, поэтому всего Игорь может использовать 80 различных кодовых слов.

Ответ: 8.

При объяснении этого класса задач обязательно разъяснять учащимся комбинаторную природу формулы a^k , где a — количество символов в алфавите (возможных сигналов

прибора), а k — количество символов в слове (сигналов в сообщении). Многие учителя, судя по всему, ограничиваются только лишь сообщением самой формулы, в результате чего распространённой ошибкой является указание в ответе значения k^a вместо a^k . Также учителю информатики полезно будет напомнить учащимся, что операция возведения числа a в степень k представляет собой k последовательных умножений a . Задание 10 в своём нынешнем виде подчёркивает комбинаторную природу этой формулы, поскольку на использование определённых букв вводятся ограничения.

С равномерным кодированием связано также **задание 9**. Оно проверяет тот же материал, только применительно к кодированию растровых изображений или графических файлов. В случае графики количество вариантов сигнала в сообщении равно количеству цветов в палитре, а длина сообщения представляет собой количество точек (пикселей) в изображении. Задание может заключаться либо в определении информационного объёма сигнала (объёма памяти, необходимого для хранения изображения без сжатия), либо в определении количества цветов в палитре.

Пример задания 9

Для хранения произвольного растрового изображения размером 1024×1024 пикселей отведено 512 Кбайт памяти, при этом для каждого пикселя хранится двоичное число — код цвета этого пикселя. Для каждого пикселя для хранения кода выделено одинаковое количество бит. Сжатие данных не производится. Какое максимальное количество цветов можно использовать в изображении?

Решение: Необходимо выяснить, сколько бит отводится на хранение одного пикселя изображения. Для этого естественно объём памяти разделить на размер изображения. Правильнее делать это с использованием свойств степенной функции. Для этого, помня, что $512 = 2^9$, а $1024 = 2^{10}$, а также, что в Кбайт 1024 байт, а в байт 8 (2^3) бит, запишем выражение в виде частного степеней

$$N = 2^9 \cdot 2^{10} \cdot 2^3 / 2^{10} \cdot 2^{10} = 2^{22} / 2^{20} = 2^2 = 4.$$

На хранение одного пикселя отводится 4 бита. Количество различных кодов, которые можно записать в 4 бита, равно 2^4 , то есть 16.

Итак, в изображении используется палитра из 16 цветов.

Ответ: 16.

Кстати, также можно рассказать учащимся про видеоадаптеры EGA с 16-цветной палитрой, чтобы у них не было ощущения, что данная задача — простое упражнение в арифметике. Стоит отметить, что разрешение экрана у этого видеоадаптера было 640×350 пикселей, то есть данное изображение можно было бы вывести на экран только частично.

Задание 5, проверяющее знание принципов неравномерного кодирования и умение строить неравномерный код, обеспечивающий минимальную длину сообщения и безошибочность декодирования, до 2016 г. имело формат задания с выбором ответа. В 2015 г. оно стояло на 1 позиции, и было выполнено с показателем 43,5%, что для первого задания в варианте, конечно, недостаточно. В 2016 г. показатель выполнения этого задания снизился до 41% (при том, что результат выполнения экзамена в целом повысился). Видимо, участники экзамена оказались не готовы к новому содержанию заданий на эту тему.

Неравномерным называется способ кодирования, при котором разные символы алфавита кодируются битовой последовательностью различной длины. Этот код может оказаться более экономным в случае, если определённые символы алфавита встречаются в тексте часто, а некоторые — достаточно редко. Тогда часто встречающиеся символы следует кодировать короткими кодовыми последовательностями, а редко встречающиеся — длинными. На практике чаще используется равномерное кодирование, но для некоторых видов алгоритмов знание принципов неравномерного кодирования может оказаться полезным.

При решении задач на неравномерное кодирование необходимо знать, когда код является однозначно декодируемым. Для этого введём понятие префиксного кода.

Префиксным называется код, не имеющий ни одного кодового слова, которое было бы префиксом (началом) любого другого кодового слова данного кода. Если код является префиксным, то любая последова-

тельность кодовых слов всегда только единственным образом разделяема на отдельные из них. Требование префиксного кода является условием Фано, которое было явно сформулировано в задании 5 ряда вариантов 2016 года.

Сначала разберём задание 5 из демоверсии 2016 года, а потом задание одного из вариантов.

Задание 5.1

По каналу связи передаются сообщения, содержащие только четыре буквы: П, О, С, Т; для передачи используется двоичный код, допускающий однозначное декодирование. Для букв Т, О, П используются такие кодовые слова: Т: 111, О: 0, П: 100. Укажите кратчайшее кодовое слово для буквы С, при котором код будет допускать однозначное декодирование. Если таких кодов несколько, укажите код с наименьшим числовым значением.

Решение: Код для буквы С должен быть трёхзначным и начинаться на 1, так как при коде 01 строки СОО и ОП будут кодироваться одинаковой последовательностью 0100, при коде 10 будут совпадать СО и П, при коде 11 совпадут последовательности ТП и ССО. Трёхзначный код 101 обеспечит однозначное декодирование.

Ответ: 16.

Это простое переборное решение, в котором в неявном виде реализован общий подход к решению подобных задач с помощью построения дерева возможных кодов. Разберём этот общий метод на примере одного из заданий 2016 года.

Задание 5.2

Для кодирования некоторой последовательности, состоящей из букв А, Б, В, Г, Д, Е, решили использовать неравномерный двоичный код, удовлетворяющий условию Фано. Для буквы А использовали кодовое слово 0; для буквы Б — кодовое слово 10. Какова наименьшая возможная сумма длин всех шести кодовых слов?

Примечание. Условие Фано означает, что никакое кодовое слово не является началом другого кодового слова. Это обеспечивает возможность однозначной расшифровки закодированных сообщений.

Решение: Требуется построить код, отвечающий условию Фано. Проще всего это делать, рисуя двоичное дерево возможных кодов. Ясно, что код любой из оставшихся 4 букв будет начинаться на 11, так как любой код, начинающийся на 0 или на 10, не будет удовлетворять условию Фано. Для кодирования 4 символов можно использовать равномерный код длины 4: 1100, 1101, 1110, 1111. Можно для одного из символов использовать трёхбитный код, например 110. Тогда длины трёх оставшихся кодов будут, соответственно, 4, 5 и 5 (1110, 11110 и 11111). В первом случае сумма длин шести слов равна 19 бит, во втором случае $1 + 2 + 3 + 4 + 5 + 5 = 20$, что на 1 бит больше.

Ответ: 19.

Кстати, если использовать для кодирования 6 символов равномерный код, то можно обойтись длиной в 3 бита для каждого кодового слова, то есть тогда сумма длин всех 6 слов будет 18, при этом останется 2 неиспользуемых кодовых слова той же длины. Данный неравномерный код окажется экономным только в том случае, если сигнал преимущественно будет состоять из букв А и Б. Это применимо, например, к работе охранной системы, которая в обычном режиме подаёт сигнал о своей исправности и отсутствии угроз (сигнал А), в случае своей неисправности (отключения) подаёт сигнал Б, а оставшиеся 4 кода использует для обозначения сработавшего датчика сигнализации.

Данное задание оказалось неожиданно сложным даже для сильных выпускников, получивших высокие тестовые баллы. При решении этого конкретного задания верный ответ дали 41,2% экзаменуемых, ответ «20» дали 18,5% участников экзамена. Возможно, среди них как раз были сильные учащиеся, построившие по алгоритму безупречный неравномерный код, удовлетворявший условию Фано, но не увидевшие более эффективного кода.

Серьёзное затруднение в этом году, как и в прошлом, вызвало задание базового уровня 11 из раздела «Теория алгоритмов», при том, что выполняется оно методом формального исполнения алгоритма. Повторим здесь его разбор из методических рекомендаций прошлого года, так как формулировки за-

даний 2016 года были аналогичны формулировкам 2015 года.

Задание 11 проверяет владение выпускниками понятием **рекурсии** в алгоритмах и связанных с этим понятием умений и навыков. Оно как в 2015-м, так и в 2016 году было выполнено плохо (средний процент выполнения: 25,7 — 2015 г., 36,3 — 2016 г.). В 2014 году задание представляло собой фрагмент рекурсивного алгоритма, содержащего оператор вывода (печати) внутри вызываемой рекурсивной функции. Вопрос формулировался просто: какое количество символов (чисел) будет напечатано, то есть сколько раз будет выполнена команда вывода. В 2015 году эта модель задания была слегка модифицирована, вопрос формулировался в виде «Какая строка символов будет

напечатана?». В 2016 г. в ответе требовалось указать сумму напечатанных чисел. Существует решение этого задания методом формального исполнения (трассировки) алгоритма, то есть в результате репродуктивной деятельности, знакомой учащимся, хотя более простым для реализации является решение методом записи рекуррентных соотношений и построения таблицы значений (как это было в заданиях ЕГЭ на эту тему в прежние годы). Низкий показатель выполнения этого задания говорит о том, что понятие рекурсии многими учащимися в процессе обучения так и не было освоено.

В связи с тем, что это задание вызывает систематические затруднения, приведём здесь разбор двух различных образцов данного задания из вариантов КИМ ЕГЭ 2015 года.

Задание 11.1

Ниже на пяти языках программирования записана рекурсивная функция (процедура) F.

Бейсик	Python
<pre>SUB F(n) print n, IF n >= 7 THEN F(n - 3) F(n - 1) END IF END SUB</pre>	<pre>def F(n): print(n, end='') if n >= 7: F(n - 3) F(n - 1)</pre>
Алгоритмический язык	Паскаль
<pre>алг F(цел n) нач вывод n если n >= 7 то F(n - 3) F(n - 1) все кон</pre>	<pre>procedure F(n: integer); begin write(n); if n >= 7 then begin F(n - 3); F(n - 1) end end;</pre>
Си	
<pre>void F(int n) { printf("%d", n); if (n >= 7) { F(n - 3); F(n - 1); } }</pre>	

Что выведет программа при вызове F(9)? В ответе запишите последовательность выведенных цифр слитно (без пробелов).

Решение: Сначала необходимо изучить текст программы на одном из языков программирования и понять, что выполняет данная функция. Функция получает на вход одно число n , выводит его на экран, затем при условии, что $n \geq 7$, осуществляет два последовательных вызова $F(n - 3)$ и $F(n - 1)$, что приведёт к печати меньших значений n и дальнейшим рекурсивным вызовам.

Например, при данном $n = 9$ программа напечатает число 9, затем вызовет $F(6)$, то есть после числа 9 будет напечатано то, что выведет функция при вызове $F(6)$, затем произойдёт вызов $F(8)$. Упрощённо это можно записать так: $F(9) = 9, F(6), F(8)$, то есть ответ будет представлять собой последовательную запись (конкатенацию) цифры 9, ответа для $F(6)$ и ответа для $F(8)$.

Выпишем рекуррентное соотношение для общего случая:

$F(n) = n, F(n - 3), F(n - 1)$, при $n \geq 7$;

$F(n) = n$, при $n < 7$.

Далее заполним таблицу, что выведет функция при вызове для разных значений n :

n	Рекуррентное соотношение для $F(n)$	Результат вызова функции $F(n)$
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5
6	6	6
7	7, $F(4), F(6)$	746
8	8, $F(5), F(7)$	85746
9	9, $F(6), F(8)$	9685746

Например, вызов $F(7)$ приведёт к печати цифр 746, так как вызовы $F(4)$ и $F(6)$ только напечатают цифры 4 и 6 и не будут совершать никаких дальнейших рекуррентных вызовов.

Вызов $F(8)$ напечатает 8, затем сделает вызов $F(5)$, затем сделает вызов $F(7)$. Вызов $F(5)$ напечатает одну цифру 5, а вызов $F(7)$, как было определено ранее, напечатает 746, поэтому ответом для $F(8)$ будет 85746.

Наконец, $F(9)$ напечатает 9, сделает вызов $F(6)$, который напечатает 6, и сделает вызов $F(8)$, который напечатает 85746. Последовательно записав эти цифры, получим ответ для $F(9)$: 9685746.

Альтернативное решение: Это решение более трудоёмкое, чем приведённое выше, но основывается на формальном исполнении алгоритма, записанного в задании.

Взгляд на текст процедуры показывает, что печать происходит сразу после вызова, при этом печатается аргумент, с которым процедура была вызвана. После этого в алгоритме содержится условный оператор, который при условии, что алгоритм больше либо равен семи, осуществляет ещё два последовательных рекурсивных вызова той же процедуры с аргументом на 3 и на 1 меньше, чем первоначальный аргумент.

Исполним алгоритм для указанного аргумента:

Команда алгоритма	Результат исполнения	Вывод	Примечание
<u>ВЫВОД</u> n	вывод «9»	9	
<u>ЕСЛИ</u> $n \geq 7$ <u>ТО</u>	$9 \geq 7$, истина		
$F(n - 3)$	вызов $F(6)$		1 уровень рекурсии
<u>ВЫВОД</u> n	вывод «6»	6	
<u>ЕСЛИ</u> $n \geq 7$ <u>ТО</u>	$6 \geq 7$, ложь		возврат в $F(9)$

Команда алгоритма	Результат исполнения	Вывод	Примечание
F(n - 1)	вызов F(8)		1 уровень рекурсии
<u>ВЫВОД</u> n	ВЫВОД «8»	8	
<u>ЕСЛИ</u> n >= 7 <u>ТО</u>	8 ≥ 7, истина		
F(n - 3)	вызов F(5)		2 уровень рекурсии
<u>ВЫВОД</u> n	ВЫВОД «5»	5	
<u>ЕСЛИ</u> n >= 7 <u>ТО</u>	5 ≥ 7, ложь		возврат в F(8)
F(n - 1)	вызов F(7)		2 уровень рекурсии
<u>ВЫВОД</u> n	ВЫВОД «7»	7	
<u>ЕСЛИ</u> n >= 7 <u>ТО</u>	7 ≥ 7, истина		
F(n - 3)	вызов F(4)		3 уровень рекурсии
<u>ВЫВОД</u> n	ВЫВОД «4»	4	
<u>ЕСЛИ</u> n >= 7 <u>ТО</u>	4 ≥ 7, ложь		возврат в F(7)
F(n - 1)	вызов F(6)		3 уровень рекурсии
<u>ВЫВОД</u> n	ВЫВОД «6»	6	
<u>ЕСЛИ</u> n >= 7 <u>ТО</u>	6 ≥ 7, ложь		возврат в F(7)
<u>КОН</u>			возврат в F(8)
<u>КОН</u>			возврат в F(9)
<u>КОН</u>			завершение алгоритма

Ответ: 9685746.

Задание 11.2

Ниже на пяти языках программирования записаны две рекурсивные функции (процедуры): F и G.

Бейсик	Python
<pre> DECLARE SUB F(n) DECLARE SUB G(n) SUB F(n) IF n > 0 THEN G(n - 1) END SUB SUB G(n) PRINT "*" IF n > 1 THEN F(n - 2) END SUB </pre>	<pre> def F(n): if n > 0: G(n - 1) def G(n): print(«*») if n > 1: F(n - 2) </pre>

Алгоритмический язык	Паскаль
<pre> алг F(цел n) нач если n > 0 то G(n - 1) все кон </pre>	<pre> procedure F(n: integer); forward; procedure G(n: integer); forward; procedure F(n: integer); begin if n > 0 then </pre>

<pre> алг G(цел n) нач вывод "*" если n > 1 то F(n - 2) все кон </pre>	<pre> G(n - 1); end; procedure G(n: integer); begin writeln('*'); if n > 1 then F(n - 2); end; </pre>
Си	
<pre> void F(int n); void G(int n); void F(int n){ if (n > 0) G(n - 1); } void G(int n){ printf("*"); if (n > 1) F(n - 2); } </pre>	

Сколько символов «звёздочка» будет напечатано на экране при выполнении вызова F(11)?

Решение: В данной задаче используется так называемая косвенная рекурсия, когда функция F вызывает функцию G, а функция G вызывает функцию F.

Изучив текст программы, заметим, что функция $G(n)$ осуществляет вызов $F(n - 2)$, а вызов функции $F(n)$ приводит к вызову $G(n - 1)$. Таким образом, если функция $G(n)$ вызывает $F(n - 2)$, то она в свою очередь вызывает $G(n - 3)$.

Обозначим через $g(n)$ количество звёздочек, которое будет напечатано на экране, если вызвать функцию $G(n)$ из данного задания. Всегда будет напечатана хотя бы одна звёздочка, так как функция G начинается с команды вывода одной звёздочки, но если $n \geq 3$, то произойдёт рекурсивный вызов $G(n) \rightarrow F(n - 2) \rightarrow G(n - 3)$. При $n < 3$ рекурсивного вызова не будет, например, если $n = 2$, то $G(2)$ вызовет $F(0)$, а та просто завершит работу, не вызывая ничего.

Выпишем рекуррентное соотношение для $g(n)$:

$$g(n) = 1 + g(n - 3), \text{ при } n \geq 3,$$

$$g(n) = 1, \text{ при } n < 3.$$

Заполним таблицу значений функции $g(n)$:

n	Рекуррентное соотношение для $g(n)$	Значение функции $g(n)$
0	1	1
1	1	1
2	1	1
3	$1 + g(0)$	2
4	$1 + g(1)$	2

n	Рекуррентное соотношение для $g(n)$	Значение функции $g(n)$
5	$1 + g(2)$	2
6	$1 + g(3)$	3
7	$1 + g(4)$	3
8	$1 + g(5)$	3
9	$1 + g(6)$	4
10	$1 + g(7)$	4

В задаче спрашивается, сколько символов «звёздочка» будет напечатано на экране при выполнении вызова $F(11)$? Поскольку $F(11)$ вызывает $G(10)$ и ничего более не делает, то ответом будет $g(10)$, то есть 4.

Альтернативное решение (формальным исполнением): Взгляд на текст задания показывает, что печать происходит только внутри процедуры G сразу после вызова, при этом печатается один символ «звёздочка». Записаны две процедуры F и G , при этом из процедуры F осуществляется вызов $G(n - 1)$, а из процедуры G осуществляется вызов $F(n - 2)$. Рекурсия заканчивается, когда аргумент становится равным 1 (в процедуре G) или 0 (в процедуре F).

Исполним алгоритм для указанного аргумента (11):

Команда алгоритма	Результат исполнения	Вывод	Примечание
<u>если</u> $n > 0$ <u>то</u>	$11 > 0$, истина		выполняется $F(11)$
$G(n - 1)$	вызов $G(10)$		
<u>ВЫВОД</u> "*"	ВЫВОД «*»	*	выполняется $G(10)$
<u>если</u> $n > 1$ <u>то</u>	$10 > 1$, истина		
$F(n - 2)$	вызов $F(8)$		
<u>если</u> $n > 0$ <u>то</u>	$8 > 0$, истина		выполняется $F(8)$
$G(n - 1)$	вызов $G(7)$		
<u>ВЫВОД</u> "*"	ВЫВОД «*»	*	выполняется $G(7)$
<u>если</u> $n > 1$ <u>то</u>	$7 > 1$, истина		
$F(n - 2)$	вызов $F(5)$		
<u>если</u> $n > 0$ <u>то</u>	$5 > 0$, истина		выполняется $F(5)$
$G(n - 1)$	вызов $G(4)$		
<u>ВЫВОД</u> "*"	ВЫВОД «*»	*	выполняется $G(4)$
<u>если</u> $n > 1$ <u>то</u>	$4 > 1$, истина		
$F(n - 2)$	вызов $F(2)$		
<u>если</u> $n > 0$ <u>то</u>	$2 > 0$, истина		выполняется $F(2)$
$G(n - 1)$	вызов $G(1)$		
<u>ВЫВОД</u> "*"	ВЫВОД «*»	*	выполняется $G(1)$
<u>если</u> $n > 1$ <u>то</u>	$1 > 1$, ложь		
<u>кон</u>			завершение $G(1)$
<u>кон</u>			завершение $F(2)$
<u>кон</u>			завершение $G(4)$
<u>кон</u>			завершение $F(5)$
<u>кон</u>			завершение $G(7)$

Команда алгоритма	Результат исполнения	Вывод	Примечание
<u>кон</u>			завершение F(8)
<u>кон</u>			завершение G(10)
<u>кон</u>			завершение F(11)

Таким образом, символ «звёздочка» будет напечатан 4 раза.

Ответ: 4.

Как видно из приведённых решений к образцам заданий, от экзаменуемых не требовалось ничего больше, чем формально исполнить приведённый алгоритм. Видимо, проблема заключается в том, что разъяснению понятия рекурсии и механизма осуществления рекурсивного вызова было уделено недостаточно внимания. Понять и почувствовать конструкцию рекурсии не сложнее, чем конструкцию оператора присваивания $a = a + 1$, основной способ добиться этого понимания — решать больше практических заданий.

Например, можно добавить в описанные две процедуры ещё один оператор вывода. Для экономии места приведём алгоритмы на двух языках:

Алгоритмический язык	Паскаль
<pre> алг F(цел n) нач если n > 0 то G(n - 1) все вывод "+" кон алг G(цел n) нач вывод "*" если n > 1 то F(n - 2) все кон </pre>	<pre> procedure F(n: integer); forward; procedure G(n: integer); forward; procedure F(n: integer); begin if n > 0 then G(n - 1); writeln('*'); end; procedure G(n: integer); begin writeln('*'); if n > 1 then F(n - 2); end; </pre>

К данному алгоритму можно задать несколько вопросов. Первый: «Сколько символов «плюс» будет напечатано?» — достаточно прост и эквивалентен по сложности заданию 11.2; а другой: «Какая последовательность символов будет напечатана?» — требует глубокого понимания сущности рекурсии и ближе по сложности к заданию 11.1. (*Ответ:* при выполнении вызова F(11) будет напечатана строка «***++++».)

Эффективным способом организации занятий по данной теме может быть разбор заданий у доски с объяснением учителя, а затем самостоятельная работа учащихся с теми же алгоритмами, реализованными в среде программирования (например, в среде «КуМир»), когда учащиеся могут запустить эти алгоритмы с разными аргументами (например модифицированное задание 11.2 с аргументами 9 и 8) и обсудить, в чём причина совпадения или несовпадения результатов. Далее учащиеся могут попробовать модифицировать алгоритмы (например, переместить операторы вывода) и спрогнозировать результаты их выполнения, а затем проверить гипотезу.

В любом случае рекурсия относится к фундаментальным понятиям информатики, её изучение в школе необходимо и важно в плане общего образования.

Ещё одно задание базового уровня сложности, вызвавшее затруднения у участников экзамена, — **задание 12**, проверяющее знание базовых принципов организации и функционирования компьютерных сетей, адресации в сети. В этом году в части вариантов оно было с новой формулировкой, требовавшей определить наименьшее возможное количество единиц в разрядах маски сети. Разберём задание с новой формулировкой.

Задание 12.1

В терминологии сетей TCP/IP маской сети называется двоичное число, определяющее, какая часть IP-адреса узла сети относится к адресу сети, а какая — к адресу самого узла в этой сети. Обычно маска записывается по тем же правилам, что и IP-адрес, — в виде четырёх байтов, причём каждый байт записывается в виде десятичного числа. При этом в маске сначала (в старших разрядах) стоят единицы, а затем с некоторого разряда — нули. Адрес сети получается в результате применения поразрядной конъюнкции к заданным IP-адресу узла и маске.

Например, если IP-адрес узла равен 231.32.255.131, а маска равна 255.255.240.0, то адрес сети равен 231.32.240.0.

Для узла с IP-адресом 119.83.208.27 адрес сети равен 119.83.192.0. Каково наименьшее возможное количество единиц в разрядах маски?

Решение: Строго говоря, всё, что нужно знать для решения этого задания, написано в условии. IP-адрес представляет собой 32 бита, которые для удобства чтения записываются в виде значений 4 байтов в десятичной системе, разделяемых точками. Адрес в виде 16 единиц и следующих за ними 16 нулей будет записан в виде 255.255.0.0. Маска сети для приведённых в условии IP-адресов узла и сети будет иметь вид 255.255.X.0, где X — число, дающее при поразрядной конъюнкции с числом 208 число 192.

Запишем оба числа в двоичной системе: $192_{10} = 11000000_2$, $208_{10} = 11010000_2$.

Уже видно, что минимальное число, дающее при поразрядной конъюнкции с числом 208 число 192, есть число 192. Другое число, которое даст такой результат — число 224 (11100000_2). Число 240 при поразрядной конъюнкции с числом 208 даст 208, и адрес сети будет 119.83.208.0.

В двоичной записи числа 255 содержится 8 единиц, числа 192 — 2 единицы. Маска будет содержать последовательно 18 единиц и 14 нулей.

Ответ: 18.

Имеет смысл, видимо, разобрать то же задание в другой формулировке, использовавшейся на экзамене как в 2015 г., так и в части вариантов 2016 г.

Задание 12.2

В терминологии сетей TCP/IP маской сети называется двоичное число, определяющее, какая часть IP-адреса узла сети относится к адресу сети, а какая — к адресу самого узла в этой сети. Обычно маска записывается по тем же правилам, что и IP-адрес, — в виде четырёх байтов, причём каждый байт записывается в виде десятичного числа. При этом в маске сначала (в старших разрядах) стоят единицы, а затем с некоторого разряда — нули. Адрес сети получается в результате применения поразрядной конъюнкции к заданным IP-адресу узла и маске.

Например, если IP-адрес узла равен 231.32.255.131, а маска равна 255.255.240.0, то адрес сети равен 231.32.240.0.

Для узла с IP-адресом 115.181.92.48 адрес сети равен 115.181.80.0. Чему равно значение третьего слева байта маски? Ответ запишите в виде десятичного числа.

Решение: Маска сети для приведённых в условии IP-адресов узла и сети будет иметь вид 255.255.X.0, где X — число, дающее при поразрядной конъюнкции с числом 92 число 80.

Запишем оба числа в двоичной системе: $92_{10} = 01011100_2$, $80_{10} = 01010000_2$. Мы записали их с лидирующим нулём, чтобы получился 8-битный код, соответствующий байту.

Маска должна представлять собой число, в двоичной записи которого сначала идут единицы, а потом нули. Так как 4 младших разряда третьего байта адреса содержат нули, а че-

тыре старших разряда адреса сети и адреса узла совпадают, маска должна представлять собой число, в двоичной записи которого сначала 4 единицы, а потом 4 нуля. Это число 240.

Ответ: 240.

Если всерьёз, то решение первой задачи ничуть не сложнее, если не легче, чем решение второй. Причина, по которой в этом году с ней справились только 30% экзаменуемых против 40% в прошлом году, может быть только одна — испуг от необычной формулировки и нежелание вчитаться в условие. Для экзамена, который определяет возможность продолжения образования по выбранной специальности, такая реакция не вполне адекватна. Или другая возможная причина более высоких результатов прошлого года состоит в том, что часть участников экзамена не решали задачу полностью, а запомнили «магические числа» 192, 224 и 240, которые могли бы быть ответом на задание, и подбирали, по каким-то признакам, подходящее. Видимо, они не смогли перестроиться и вчитаться в условие задачи. С другой стороны, ответы «192» и «224» дали суммарно около 2% выпускников, решавших вариант, в котором было задание 12.1. Остаётся только повторить, что формулировки задания на реальном экзамене совсем не обязательно будут совпадать с формулировками в демоверсии и в заданиях открытого банка.

Задание 18 повышенного уровня сложности проверяет знание таблицы истинности для импликации и умение осуществить преобразование импликации в сложных выражениях. В 2015 г. задание в ряде вариантов заодно проверяло умение осуществить поразрядную конъюнкцию двоичных чисел. Оно вызвало серьёзные затруднения.

Задание 18.1

Обозначим через $m \& n$ поразрядную конъюнкцию неотрицательных целых чисел m и n . Так, например, $14 \& 5 = 1110_2 \& 0101_2 = 0100_2 = 4$.

Для какого наименьшего неотрицательного целого числа A формула

$$x \& 25 \neq 0 \rightarrow (x \& 17 = 0 \rightarrow x \& A \neq 0)$$

тождественно истинна (то есть принимает значение 1 при любом неотрицательном целом значении переменной x)?

Решение: Поскольку выражения $A \rightarrow (B \rightarrow C)$ и $(A \wedge B) \rightarrow C$ равносильны, выражение $x \& 25 \neq 0 \rightarrow (x \& 17 = 0 \rightarrow x \& A \neq 0)$ равносильно выражению $(x \& 25 \neq 0 \wedge x \& 17 = 0) \rightarrow x \& A \neq 0$.

Поскольку $25_{10} = 11001_2$, то $x \& 25 \neq 0$ означает, что или нулевой, или третий, или четвёртый разряд в двоичной записи числа x не равен 0.

Аналогично, $x \& 17 = 0$ означает, что нулевой и четвёртый разряд в двоичной записи числа x равны 0. Следовательно, из $(x \& 25 \neq 0 \wedge x \& 17 = 0)$ следует, что третий разряд в двоичной записи числа x не равен 0. Поэтому если $x \& A \neq 0$, то выражение

$$(x \& 25 \neq 0 \wedge x \& 17 = 0) \rightarrow x \& A \neq 0$$

истинно при любом x , если же $x \& A = 0$, то выражение ложно при, например, $x = 8$. Наименьшее A , при котором $x \& A \neq 0$ равно 8.

Ответ: 8.

В 2016 году в некоторых вариантах задание 18 было приведено в формулировке, требовавшей определения длин отрезков числовой прямой. Приведём пример задания 18 одного из вариантов 2016 года.

Задание 18.2

На числовой прямой даны два отрезка: $P = [30, 65]$ и $Q = [10, 35]$. Отрезок A таков, что формула $(x \in A) \rightarrow ((x \in P) \rightarrow (x \in Q))$ истинна при любом значении переменной x .

Какова наименьшая возможная длина отрезка A ?

Решение: Для начала можно применить преобразование импликации два раза и получить выражение без импликации:

$(x \in A) \rightarrow ((x \in P) \rightarrow (x \in Q))$ равносильно выражению $(x \in A) \rightarrow ((x \in P) \vee (x \in Q))$ (мы преобразовали импликацию в скобках и применили отрицание, заменив «принадлежит» на «не принадлежит»). Аналогично это выражение равносильно $(x \in A) \vee ((x \in P) \vee (x \in Q))$.

Скобки можно раскрыть, получаем $(x \in A) \vee (x \in P) \vee (x \in Q)$.

Для этого выражения формула истинна для всех x , не принадлежащих либо отрезку P , либо отрезку Q . Чтобы она была истинна для всей числовой прямой, требуется, чтобы отрезок A полностью покрывал пересечение отрезков P и Q . Минимальный такой отрезок (30, 35) совпадает с пересечением отрезков P и Q и имеет длину 5.

Ответ: 5.

Как видно из приведённых примеров, задание не требует для выполнения каких-то знаний и умений, выходящих за рамки стандартных логических преобразований, но требует хорошего понимания глубокой взаимосвязи операций пересечения и объединения множеств с логическими операциями конъюнкции и дизъюнкции. Эти множества могут представлять собой отрезки на прямой или натуральные числа, имеющие те или иные цифры в двоичном разложении.

Модель КИМ 2017 г. по сравнению с КИМ 2016 года не изменится. Останутся теми же, что и в 2015–2016 гг., количество заданий и максимальный первичный балл. Задание 27 будет, как и в 2016 году, представлено в двух вариантах, оцениваемых, соответственно, из 2 и 4 баллов. Для написания программ в задании 27 можно будет использовать любой изучавшийся язык программирования, в том числе стандартные библиотеки. Проверка решений заданий второй части будет осуществляться экспертами по критериям, использование компьютера на этапах выполнения заданий и ввода ответов не предусмотрено.

Несмотря на то, что в КИМ ЕГЭ 2017 года не будет заданий с выбором ответа, не следует игнорировать задания открытого банка ЕГЭ этого типа. Решение заданий ЕГЭ прошлых лет поможет учащимся представить разнообразие сюжетов и проверяемых элементов содержания по каждому из разделов содержания.