

ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ И НАУКИ ГОРОДА МОСКВЫ
Государственное автономное образовательное учреждение
дополнительного профессионального образования города Москвы
«МОСКОВСКИЙ ЦЕНТР КАЧЕСТВА ОБРАЗОВАНИЯ»

УПРАВЛЕНИЕ РАЗВИТИЯ ЦИФРОВЫХ, ИНТЕРАКТИВНЫХ И
ДИСТАНЦИОННЫХ ТЕХНОЛОГИЙ ОБРАЗОВАНИЯ

СОГЛАСОВАНО

Начальник управления
государственного надзора и контроля в
сфере образования Департамента
образования и науки города Москвы



И.В. Гуськов

« ____ » _____ 2019 г.



УТВЕРЖДАЮ

Зам. директора ГАОУ ДПО МЦКО
П.Л. Лепе

« ____ » _____ 2019 г.

**Дополнительная профессиональная программа
(повышение квалификации)**

**Подготовка экспертов для работы в предметной комиссии при проведении
государственной итоговой аттестации по образовательным программам среднего
общего образования (ГИА-11) по информатике в 2020 году**
(для слушателей с опытом работы в предметной комиссии не более 2-х лет)

Разработчик курса:
Лещинер В.Р.

Рег. номер 13

Москва – 2019

Раздел 1. «Характеристика программы»

1.1. Цель реализации программы

Формирование профессиональных компетенций слушателей в области проверки и оценки выполнения заданий с развернутым ответом экзаменационных работ ГИА-11 по информатике.

Формируемые компетенции

№ п/п	Компетенции	Направление подготовки Педагогическое образование
		Квалификация Магистратура
		Код компетенции 44.04.01
1.	Способен осуществлять и оптимизировать профессиональную деятельность в соответствии с нормативными правовыми актами в сфере образования и нормами профессиональной этики	ОПК-1

1.2. Планируемые результаты обучения

№ п/п	Знать – уметь	Направление подготовки Педагогическое образование
		Квалификация Магистратура
		Код компетенции 44.04.01
1.	Знать: – полномочия и функции членов предметной комиссии; – содержание нормативных документов, определяющих структуру и содержание контрольных измерительных материалов для проведения государственной итоговой аттестации (ГИА-11); – типологию заданий с развернутым ответом, используемых в КИМ ЕГЭ и ГВЭ по информатике; – технологии объективной оценки образовательных достижений и критерии оценивания экзаменационных работ; порядок проверки и оценки ответов участников экзамена на задания с развернутым ответом по информатике. Уметь: – работать с инструкциями, регламентирующими процедуру проверки и оценки ответов участников экзамена на задания с развернутым ответом; – проверять и объективно оценивать ответы	ОПК-1

	участников экзамена на задания с развернутым ответом; – оформлять результаты проверки, соблюдая установленные требования.	
--	--	--

1.3. Категория обучающихся: уровень образования – ВО, направление подготовки – «Педагогическое образование», область профессиональной деятельности – обучение информатике; на уровне среднего общего образования.

Программа для слушателей без опыта работы в ПК / с опытом работы в ПК не более 2-х лет.

1.4. Форма обучения: очная с дистанционной поддержкой обучения.

1.5. Режим занятий, срок освоения программы: 4-6 часов в день.

1.6. Трудоемкость программы: 54 часа.

Раздел 2. «Содержание программы»

2.1. Учебный (тематический) план

№ п/ п	Наименование разделов (модулей) и тем	Аудиторные учебные занятия, учебные работы				Форма контро ля
		Всего ауд. час	Лекции	Практи- ческие занятия	Самосто- ятельная работа	
1.	Теоретическая часть	10	2	8		
1.1	Нормативные правовые основы проведения ГИА-11	2	2			
1.2	Структура и содержание КИМ ГИА-11 по информатике	4		4		
1.3	Специфика оценки заданий с развернутым ответом по информатике	4		4		
2.	Практическая часть	38		28	10	
2.1	Анализ результатов работы ПК ГИА-11 по информатике	2		2		
2.2	Практикум по оцениванию заданий с развернутым ответом по информатике	30		20	10	Практи- ческие работы №№ 1- 5
2.3	Анализ согласованности проверки заданий с развернутым ответом по итогам практических работ	6		6		
3.	Итоговая аттестация	6		6		Зачет. Тестир ование
	Итого:	54	2	42	10	

2.2. Учебная программа

№ п/п	Виды учебных занятий, учебных работ	Содержание
1. Теоретическая часть		
Тема 1.1 Нормативные правовые основы проведения ГИА-11	Лекция (2 ч.)	Нормативная правовая база при подготовке и проведении ГИА по программам среднего общего образования: федеральные, ведомственные и региональные, порядок проведения ГИА-11. Организация работы предметной комиссии ГИА-11: инструктивно-методические материалы и регламент работы ПК ГИА-11 на ППЗ. Квалификационные характеристики экспертов. Статусы экспертов ПК ГИА. Формирование и организация работы ПК. Требования к порядку работы эксперта ПК. Соблюдение Российского законодательства, соответствие принципам гуманизма и целесообразности. Профессионально-личностные качества эксперта.
Тема 1.2 Структура и содержание КИМ по информатике	Практическое занятие (4 ч.)	Структура и содержание КИМ ГИА-11 в 2020 году по информатике: кодификатор элементов содержания и требований к уровню подготовки обучающихся, спецификация КИМ, демонстрационный вариант КИМ ГИА-11 в 2020 году. Типы заданий ГИА-11 по информатике, распределение заданий экзаменационной работы по проверяемым контрольным элементам содержания (КЭС). Структура и содержание КИМ ГВЭ в 2020 г.
Тема 1.3 Специфика оценки заданий с развернутым ответом по информатике	Практическое занятие (4 ч.)	Система оценки выполнения заданий с развернутым ответом ЕГЭ по информатике. Виды шкал, используемых для оценки выполнения каждого типа заданий с развернутым ответом по информатике. Процедура проверки и оценки заданий с развернутым ответом в рамках проведения ЕГЭ: протокол проверки, методика назначения третьего эксперта. Процедура проверки и оценки заданий с развернутым ответом в рамках проведения ГВЭ: протокол проверки, методика назначения третьего эксперта.
2. Практическая часть		
Тема 2.1 Анализ результатов	Практическое занятие	Итоги экзамена по информатике в 2019 г.: анализ результатов работы предметной

работы ПК ГИА-11 по информатике	(2 ч.)	комиссии. Преодоление расхождения баллов в работе экспертов.
Тема 2.2 Практикум по оцениванию заданий с развернутым ответом по информатике	Практическое занятие (4 ч.)	Методические указания ФГБНУ ФИПИ по оцениванию заданий с развернутым ответом по информатике и ИКТ. Сложные случаи оценивания отдельных заданий. Согласование критериев оценивания в целях единообразия оценивания работ, выполненных на различных языках программирования.
	Самостоятельная работа (2 ч.)	Выполнение практической работы № 1 (отработка критериального оценивания задания № 24)
	Практическое занятие (4 ч.)	Проверка заданий 24 и 25. Критерии оценивания. Обоснование результатов проверки, выделение ошибок участников экзамена, аргументация и пояснение позиции эксперта. Способы разрешения проблемных ситуаций при проверке ответов участников экзамена.
	Самостоятельная работа (2 ч.)	Выполнение практической работы № 2 (отработка критериального оценивания задания № 25)
	Практическое занятие (4 ч.)	Проверка задания 26. Критерии оценивания. Обоснование результатов проверки, выделение ошибок участников экзамена, аргументация и пояснение позиции эксперта. Способы разрешения проблемных ситуаций при проверке ответов участников экзамена.
	Самостоятельная работа (2 ч.)	Выполнение практической работы № 3 (отработка критериального оценивания задания № 26)
	Практическое занятие (4 ч.)	Проверка задания 27. Критерии оценивания. Обоснование результатов проверки, выделение ошибок участников экзамена, аргументация и пояснение позиции эксперта. Способы разрешения проблемных ситуаций при проверке ответов участников экзамена.
	Самостоятельная работа (2 ч.)	Выполнение практической работы № 4 (отработка критериального оценивания задания № 27)

Тема 2.3 Анализ согласованности проверки заданий с развернутым ответом по итогам практических работ	Практическое занятие (4 ч.)	Проверка экзаменационной работы в целом. Механизмы проверки. Сквозная проверка и проверка по заданиям: преимущества и недостатки каждого метода. Обоснование результатов проверки, выделение ошибок участников ГИА, аргументация и пояснение позиции эксперта. Способы разрешения проблемных ситуаций при проверке ответов участников экзамена.
	Самостоятельная работа (2 ч.)	Выполнение практической работы № 5 (отработка критериального оценивания работы в целом)
	Практическое занятие (6 ч.)	Обобщение результатов практических работ по проверке заданий с развернутым ответом. Проблемные ситуации при оценивании. Способы достижения единых подходов к проверке и оценке отдельных заданий и работы в целом. Порядок прохождения квалификационных испытаний на присвоение статуса эксперта ПК ГИА.
3. Итоговая аттестация	6 ч.	Зачет Тестирование

Раздел 3. «Формы аттестации и оценочные материалы»

3.1. Текущий контроль: практические работы №№ 1-5 – проверка и критериальное оценивание ответов на задания с развернутым ответом, оформление результатов проверки.

Материалы практических работ размещены в ИОС образовательной организации.

Практические работы №№ 1-5.

Практическая работа №1

Текст задания

Оцените согласно критериям оценивания выполнение задания 24 КИМ ЕГЭ по информатике. Оценки выставьте в протокол оценивания:

На обработку поступает натуральное число, не превышающее 10^9 . Нужно написать программу, которая выводит на экран количество цифр этого числа, кратных 5. Если в числе нет цифр, кратных 5, требуется на экран вывести «NO». Программист написал программу неправильно. Ниже эта программа для Вашего удобства приведена на пяти языках программирования.

Напоминание: 0 делится на любое натуральное число.

Бейсик	Python
<pre> DIM N, DIGIT, COUNT AS LONG INPUT N COUNT = 1 WHILE N > 0 DIGIT = N MOD 10 IF DIGIT MOD 5 = 0 THEN COUNT = COUNT + DIGIT END IF N = N \ 10 WEND IF COUNT = 0 THEN PRINT "NO" ELSE PRINT COUNT END IF </pre>	<pre> N = int(input()) count = 1 while N > 0: digit = N % 10 if digit % 5 == 0: count = count + digit N = N // 10 if count == 0: print("NO") else: print(count) </pre>
Алгоритмический язык	Паскаль
<pre> алг нач цел N, digit, count ввод N count := 1 нц пока N > 0 digit := mod(N,10) если mod(digit,5) = 0 то count := count + digit все N := div(N,10) кц если count = 0 то вывод "NO" иначе вывод count все кон </pre>	<pre> var N, digit, count: longint; begin readln(N); count := 1; while N > 0 do begin digit := N mod 10; if digit mod 5 = 0 then count := count + digit; N := N div 10; end; if count = 0 then writeln('NO') else writeln(count) end. </pre>

Си

```
#include <stdio.h>
int main()
{
    int N, digit, count;
    scanf("%d", &N);
    count = 1;
    while (N > 0)
    {
        digit = N % 10;
        if (digit % 5 == 0)
            count = count + digit;
        N = N / 10;
    }
    if (count == 0)
        printf("NO");
    else
        printf("%d", count);
    return 0;
}
```

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 164.
2. Приведите пример такого трёхзначного числа, при вводе которого программа выдаёт верный ответ.
3. Найдите все ошибки в этой программе (их может быть одна или несколько). Известно, что каждая ошибка затрагивает только одну строку и может быть исправлена без изменения других строк. Для каждой ошибки:

1) выпишите строку, в которой сделана ошибка;

2) укажите, как исправить ошибку, т.е. приведите правильный вариант строки.

Достаточно указать ошибки и способ их исправления для одного языка программирования.

Обратите внимание, что требуется найти ошибки в имеющейся программе, а не написать свою, возможно, использующую другой алгоритм решения. Исправление ошибки должно затрагивать только строку, в которой находится ошибка.

Критерии оценивания

Содержание верного ответа и указания по оцениванию
(допускаются иные формулировки ответа, не искажающие его смысла)

Решение использует запись программы на Паскале. Допускается использование программы на любом из четырёх других языков. 1. Программа выведет число 1. 2. Программа выдаёт правильный ответ, например, для числа 160. <i>Замечание для проверяющего. Программа работает неправильно из-за неверного задания начального значения счётчика и неверного увеличения счётчика. Соответственно, программа будет работать верно, если в числе есть ровно один значащий 0 и нет других цифр, кратных 5.</i> 3. В программе есть две ошибки. Первая ошибка. Неверное начальное значение счётчика. Строка с ошибкой: <pre>count := 1;</pre> Верное исправление: <pre>count := 0;</pre> Вторая ошибка. Неверное изменение счётчика. Строка с ошибкой: <pre>count := count + digit;</pre> Верное исправление: <pre>count := count + 1;</pre>	
Указания по оцениванию	Баллы
Обратите внимание! В задаче требовалось выполнить четыре действия: 1) указать, что выведет программа при конкретных входных данных; 2) указать пример подаваемого на вход числа, для которого программа выведет верный результат; 3) исправить первую ошибку; 4) исправить вторую ошибку. Для проверки правильности выполнения п. 2) нужно формально выполнить исходную (ошибочную) программу с входными данными, которые указал экзаменуемый, и убедиться в том, что результат, выданный программой, будет таким же, как и для правильной программы. Для действий 3) и 4) ошибка считается исправленной, если выполнены оба следующих условия: а) правильно указана строка с ошибкой; б) указан такой новый вариант строки, что при исправлении другой ошибки получается правильная программа.	
Выполнены все четыре необходимых действия, и ни одна верная строка не указана в качестве ошибочной	3
Не выполнены условия, позволяющие поставить 3 балла. Имеет место одна из следующих ситуаций: а) выполнены три из четырёх необходимых действий. Ни одна верная строка не указана в качестве ошибочной; б) выполнены все четыре необходимых действия. Указано в качестве ошибочной не более одной верной строки	2
Не выполнены условия, позволяющие поставить 2 или 3 балла. Выполнены два необходимых действия из четырёх	1
Не выполнены условия, позволяющие поставить 1, 2 или 3 балла.	0
Максимальный балл	3

Образец ответа участника экзамена

N 24

1. Выведет : 1 .

2. Таких чисел не существует, т.к. программа зависит от количества цифр кратных пяти и их суммы. Если в числе одна цифра кратная 5, то переменная count будет равна ~~к~~ $1 + 5 = 6$.

3. 1) count := 1; Изменение: count := 0;

2) count := count + digit; Изменение: count := count + 1;

Эталонная оценка для данного ответа: 2 (верно выполнено 3 действия из 4)

Практическая работа №2

Текст задания

Оцените согласно критериям оценивания выполнение задания 25 КИМ ЕГЭ по информатике. Оценки выставьте в протокол оценивания:

Дан целочисленный массив из 20 элементов. Элементы массива могут принимать целые значения от $-10\,000$ до $10\,000$ включительно. Опишите на естественном языке или на одном из языков программирования алгоритм, позволяющий найти и вывести количество пар элементов массива, в которых хотя бы одно число делится на 13. В данной задаче под парой подразумевается два подряд идущих элемента массива.

Например, для массива из пяти элементов: 6; 2; 13; -26; 14 – ответ: 3.

Исходные данные объявлены так, как показано ниже на примерах для некоторых языков программирования и естественного языка. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать некоторые из описанных переменных.

Бейсик	Python
<pre>CONST N AS INTEGER = 20 DIM A (1 TO N) AS INTEGER DIM I AS INTEGER, J AS INTEGER, K AS INTEGER FOR I = 1 TO N INPUT A(I) NEXT I ... END</pre>	<pre># допускается также # использовать две # целочисленные переменные j и k a = [] n = 20 for i in range(0, n): a.append(int(input())) ...</pre>
Алгоритмический язык	Паскаль

<u>алг</u> <u>нач</u> <u>цел</u> N = 20 <u>целтаб</u> a[1:N] <u>цел</u> i, j, k <u>нц</u> <u>для</u> i <u>от</u> 1 <u>до</u> N <u>ввод</u> a[i] <u>кц</u> ... <u>кон</u>	const N = 20; var a: array [1..N] of integer; i, j, k: integer; begin for i := 1 to N do readln(a[i]); ... end.
Си	Естественный язык
<pre>#include <stdio.h> #define N 20 int main() { int a[N]; int i, j, k; for (i = 0; i<N; i++) scanf("%d", &a[i]); ... return 0; }</pre>	Объявляем массив A из 20 элементов. Объявляем целочисленные переменные I, J, K. В цикле от 1 до 20 вводим элементы массива A с 1-го по 20-й. ...

В качестве ответа Вам необходимо привести фрагмент программы (или описание алгоритма на естественном языке), который должен находиться на месте многоточия. Вы можете записать решение также на другом языке программирования (укажите название и используемую версию языка программирования, например Free Pascal 2.6) или в виде блок-схемы. В этом случае Вы должны использовать те же самые исходные данные и переменные, какие были предложены в условии (например, в образце, записанном на естественном языке).

Критерии оценивания

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки решений, приводящие к правильному результату)
На языке Паскаль
<pre>k := 0; for i := 1 to N-1 do if (a[i] mod 13=0) or (a[i+1] mod 13=0) then inc(k); writeln(k);</pre>
На алгоритмическом языке
<pre>k := 0; нц для i от 1 до N-1 если mod(a[i],13)=0 или mod(a[i+1],13)=0 то k := k+1 все</pre>

КЦ ВЫВОД k	
На языке Бейсик	
<pre> K = 0 FOR I = 1 TO N-1 IF (A(I) MOD 13 = 0) OR (A(I + 1) MOD 13 = 0) THEN K = K+1 END IF NEXT I PRINT K </pre>	
На языке Си	
<pre> k = 0; for (i = 0; i<N-1; i++) if (a[i]%13 == 0 a[i+1]%13 == 0) k++; printf("%d", k); </pre>	
На языке Python	
<pre> k = 0 for i in range(0, n - 1): if (a[i] % 13 == 0 or a[i + 1] % 13 == 0): k += 1 print(k) </pre>	
На естественном языке	
Записываем в переменную <i>K</i> начальное значение, равное 0. В цикле от первого элемента до предпоследнего находим остаток от деления текущего и следующего элемента массива на 13. Если хотя бы один остаток равен 0, увеличиваем переменную <i>K</i> на единицу. После завершения цикла выводим значение переменной <i>K</i>	
Указания по оцениванию	Баллы
<i>Общие указания.</i> 1. В алгоритме, записанном на языке программирования, допускается наличие отдельных синтаксических ошибок, не искажающих замысла автора программы. 2. Эффективность алгоритма не имеет значения и не оценивается. 3. Допускается запись алгоритма на языке программирования, отличном от языков, перечисленных в условии. В этом случае должны использоваться переменные, аналогичные описанным в условии. Если язык программирования использует типизированные переменные, описания переменных должны быть аналогичны описаниям переменных на естественном языке. Использование нетипизированных или необъявленных переменных возможно только в случае, если это допускается языком программирования; при этом количество переменных и их идентификаторы должны соответствовать условию задачи	

Предложен правильный алгоритм, выдающий в качестве результата верное значение	2
Не выполнены условия, позволяющие поставить 2 балла. Предложено в целом верное решение, содержащее не более одной ошибки из числа следующих: 1) в цикле происходит выход за границу массива (например, используется цикл от 1 до N); 2) не инициализируется или неверно инициализируется счётчик количества найденных пар; 3) счётчик количества пар в цикле не изменяется или изменяется неверно; 4) неверно проверяется делимость на 13; 5) на делимость проверяются не сами элементы, а их индексы; 6) при проверке выполнения условия для пары элементов используются неверные индексы; 7) в сложном логическом условии простые проверки верны, но условие в целом построено неверно (например, перепутаны операции «И» и «ИЛИ», неверно расставлены скобки в логическом выражении); 8) отсутствует вывод ответа; 9) используется переменная, не объявленная в разделе описания переменных; 10) не указано или неверно указано условие завершения цикла; 11) индексная переменная в цикле не меняется (например, в цикле while) или меняется неверно; 12) неверно расставлены операторные скобки	1
Ошибок, перечисленных в п. 1–12, две или больше, или алгоритм сформулирован неверно (в том числе при отсутствии цикла в явном или неявном виде)	0
<i>Максимальный балл</i>	2

Образец ответа участника экзамена

```

For i:=1 to N-1 do
  if (A[i] mod 13 = 0) or (A[i+1] mod 13 = 0) then k:=k+1;

```

Эталонная оценка для данного ответа: 0 (отсутствует инициализация счетчика и вывод результата)

Практическая работа №3

Текст задания

Оцените согласно критериям оценивания выполнение задания 26 КИМ ЕГЭ по информатике. Оценки выставьте в протокол оценивания:

Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежат две кучи камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может добавить в одну из куч (по своему выбору) **один** камень или увеличить количество камней в куче в **два раза**. Например, пусть в одной куче 10 камней, а в другой 7 камней; такую позицию в игре будем обозначать (10, 7). Тогда за один ход можно получить любую из четырёх позиций: (11, 7), (20, 7), (10, 8), (10, 14). Для того чтобы делать ходы, у каждого игрока есть неограниченное количество камней.

Игра завершается в тот момент, когда суммарное количество камней в кучах становится не менее 77. Победителем считается игрок, сделавший последний ход, т.е. первым получивший такую позицию, что в кучах всего будет 77 или больше камней.

Будем говорить, что игрок имеет *выигрышную стратегию*, если он может выиграть при любых ходах противника. Описать стратегию игрока – значит описать, какой ход он должен сделать в любой ситуации, которая ему может встретиться при различной игре противника. Например, при начальных позициях (6, 36), (7, 35), (9, 34) выигрышная стратегия есть у Пети. Чтобы выиграть, ему достаточно удвоить количество камней во второй куче.

Задание 1. Для каждой из начальных позиций (6, 35), (8, 34) укажите, кто из игроков имеет выигрышную стратегию. В каждом случае опишите выигрышную стратегию; объясните, почему эта стратегия ведёт к выигрышу, и укажите, какое наибольшее количество ходов может потребоваться победителю для выигрыша при этой стратегии.

Задание 2. Для каждой из начальных позиций (6, 34), (7, 34), (8, 33) укажите, кто из игроков имеет выигрышную стратегию. В каждом случае опишите выигрышную стратегию; объясните, почему эта стратегия ведёт к выигрышу, и укажите, какое наибольшее количество ходов может потребоваться победителю для выигрыша при этой стратегии.

Задание 3. Для начальной позиции (7, 33) укажите, кто из игроков имеет выигрышную стратегию. Опишите выигрышную стратегию; объясните, почему эта стратегия ведёт к выигрышу, и укажите, какое наибольшее количество ходов может потребоваться победителю для выигрыша при этой стратегии. Постройте дерево всех партий, возможных при указанной Вами выигрышной стратегии. Представьте дерево в виде рисунка или таблицы.

Критерии оценивания

Содержание верного ответа и указания по оцениванию (допускаются иные формулировки ответа, не искажающие его смысла)
Задание 1. В начальных позициях (6, 35), (8, 34) выигрышная стратегия

есть у Вани. При начальной позиции (6, 35) после первого хода Пети может получиться одна из следующих четырёх позиций: (7, 35), (12, 35), (6, 36), (6, 70). Каждая из этих позиций содержит менее 77 камней. При этом из любой из этих позиций Ваня может получить позицию, содержащую не менее 77 камней, удвоив количество камней во второй куче. Для позиции (8, 34) после первого хода Пети может получиться одна из следующих четырёх позиций: (9, 34), (16, 34), (8, 35), (8, 68). Каждая из этих позиций содержит менее 77 камней. При этом из любой из этих позиций Ваня может получить позицию, содержащую не менее 77 камней, удвоив количество камней во второй куче. Таким образом, Ваня при любом ходе Пети выигрывает своим первым ходом.

Задание 2. В начальных позициях (6, 34), (7, 34) и (8, 33) выигрышная стратегия есть у Пети. При начальной позиции (6, 34) он должен первым ходом получить позицию (6, 35), из начальных позиций (7, 34) и (8, 33) Петя после первого хода должен получить позицию (8, 34). Позиции (6, 35) и (8, 34) рассмотрены при разборе задания 1. В этих позициях выигрышная стратегия есть у игрока, который будет ходить вторым (теперь это Петя). Эта стратегия описана при разборе задания 1. Таким образом, Петя при любой игре Вани выигрывает своим вторым ходом.

Задание 3. В начальной позиции (7, 33) выигрышная стратегия есть у Вани. После первого хода Пети может возникнуть одна из четырёх позиций: (8, 33), (7, 34), (14, 33) и (7, 66). В позициях (14, 33) и (7, 66) Ваня может выиграть одним ходом, удвоив количество камней во второй куче. Позиции (8, 33) и (7, 34) были рассмотрены при разборе задания 2. В этих позициях у игрока, который должен сделать ход (теперь это Ваня), есть выигрышная стратегия. Эта стратегия описана при разборе задания 2. Таким образом, в зависимости от игры Пети Ваня выигрывает на первом или на втором ходу.

Примечание для эксперта. Последняя фраза в приведённом решении избыточна. Не будет ошибкой, если экзаменуемый просто напишет, например, «При выбранной стратегии партия длится не более двух ходов».

В таблице изображено дерево возможных партий при описанной стратегии Вани. Заключительные позиции (в них выигрывает Ваня) выделены жирным шрифтом.

Положения после очередных ходов				
Исходное положение	1-й ход Пети (разобраны все ходы, указана полученная позиция)	1-й ход Вани (только ход по стратегии, указана полученная позиция)	2-й ход Пети (разобраны все ходы, указана полученная позиция)	2-й ход Вани (только ход по стратегии, указана полученная позиция)
(7, 33) Всего: 40	(7, 33+1) = (7, 34) Всего: 41	(7+1, 34) = (8, 34) Всего: 42	(8+1, 34) = (9, 34) Всего: 43	(9, 34*2) = (9, 68) Всего: 77
			(8, 34+1) = (8, 35) Всего: 43	(8, 35*2) = (8, 70) Всего: 78
			(8*2, 34) = (16, 34) Всего: 50	(16, 34*2) = (16, 68) Всего: 84
			(8, 34*2) = (8, 68) Всего: 76	(8, 68*2) = (8, 136) Всего: 144
	(7+1, 33) = (8, 33) Всего: 41	(8, 33+1) = (8, 34) Всего: 42	(8+1, 34) = (9, 34) Всего: 43	(9, 34*2) = (9, 68) Всего: 77
			(8, 34+1) = (8, 35) Всего: 43	(8, 35*2) = (8, 70) Всего: 78
			(8*2, 34) = (16, 34) Всего: 50	(16, 34*2) = (16, 68) Всего: 84
			(8, 34*2) = (8, 68) Всего: 76	(8, 68*2) = (8, 136) Всего: 144
	(7*2, 33) = (14, 33) Всего: 47	(14, 33*2) = (14, 66) Всего: 80		
	(7, 33*2) = (7, 66) Всего: 73	(7, 66*2) = (7, 132) Всего: 139		

Примечание для эксперта. Дерево всех партий может быть также изображено в виде ориентированного графа – так, как показано на рисунке, или другим способом. Например, вершины дерева, соответствующие одной и той же позиции, на рисунке могут быть «склеены». Важно, чтобы множество

полных путей в графе находилось во взаимно однозначном соответствии с множеством партий, возможных при описанной в решении стратегии.

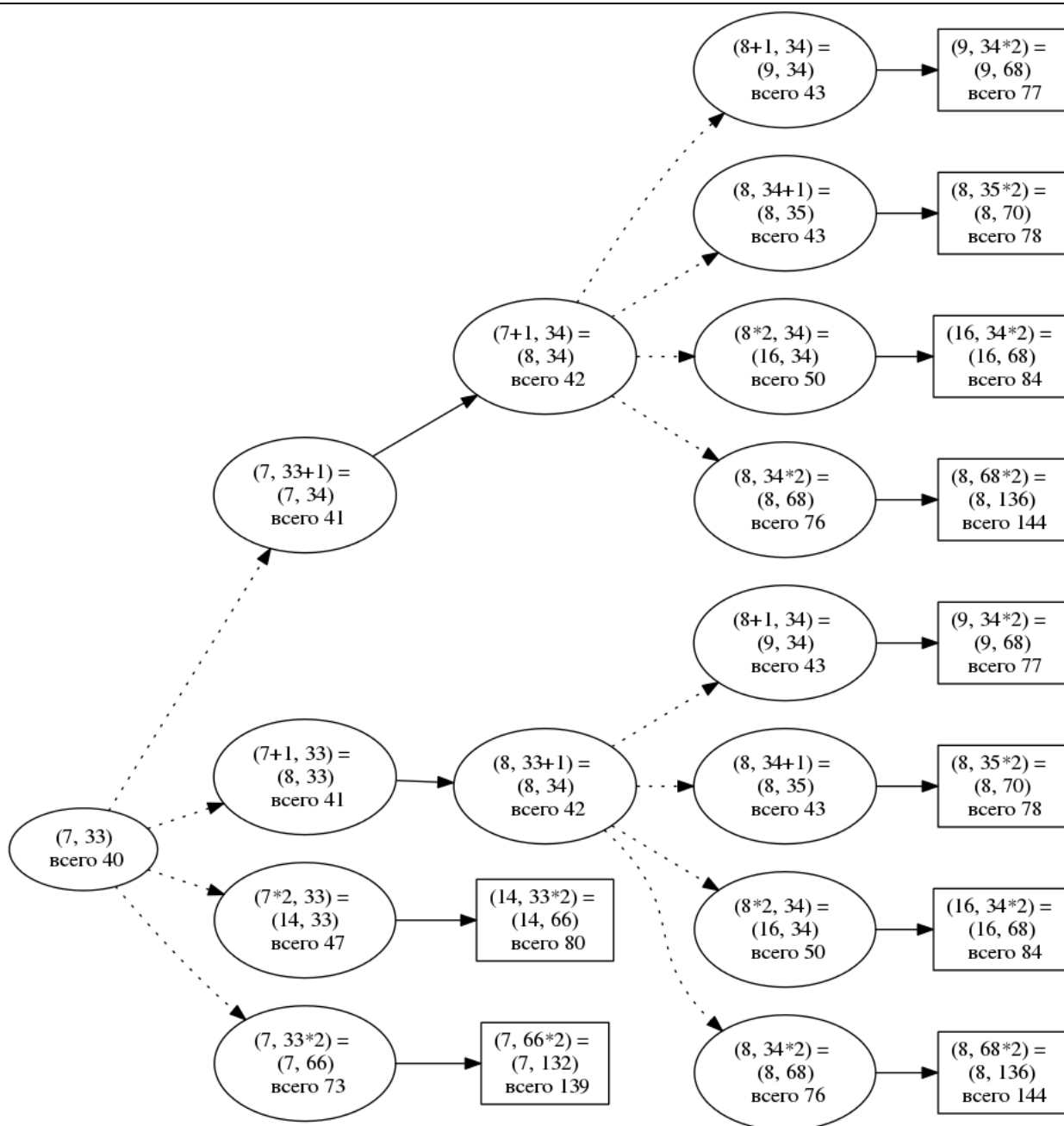


Рис. 1. Дерево всех партий, возможных при описанной стратегии Вани. Ходы Пети показаны пунктирными стрелками, ходы Вани показаны сплошными стрелками. Заключительные позиции обозначены прямоугольником

Примечание для эксперта. В некоторых позициях у Вани есть и другой способ выигрыша: например, в позиции (8, 68) можно добавить один камень в любую кучу. То, что это не указано, не является ошибкой. Экзаменуемый не должен указывать все возможные выигрышные стратегии

Указания по оцениванию	Баллы
------------------------	-------

В задаче от ученика требуется выполнить три задания. Количество баллов в целом соответствует количеству выполненных заданий (подробнее см. ниже).	
Ошибка в решении, не искажающая основного замысла и не приведшая к неверному ответу, например арифметическая ошибка при вычислении количества камней в заключительной позиции, при оценке решения не учитывается.	
Во всех случаях стратегии могут быть описаны так, как это сделано в примере решения, или другим способом	
Выполнены все три задания.	3
Здесь и далее в решениях допускаются арифметические ошибки, которые не искажают сути решения и не приводят к неправильному ответу	
Не выполнены условия, позволяющие поставить 3 балла, и выполнено хотя бы одно из следующих условий. – Выполнено задание 3. – Выполнены задания 1 и 2	2
Не выполнены условия, позволяющие поставить 2 или 3 балла, и выполнено хотя бы одно из следующих условий. – Выполнено задание 1. – Выполнено задание 2	1
Не выполнено ни одно из условий, позволяющих поставить 1, 2 или 3 балла	0
Максимальный балл	3

Образец ответа участника экзамена.

№26

1) а) $(6; 35)$ $\xrightarrow{\text{П}}$ $(6; 70)$ $\xrightarrow{\text{В}}$ $(7; 70)$

$\searrow$ $(6; 36)$ $\xrightarrow{\text{В}}$ $(6; 72)$

$\searrow$ $(7; 35)$ $\xrightarrow{\text{В}}$ $(7; 70)$

$\searrow$ $(12; 35)$ $\xrightarrow{\text{В}}$ $(12; 70)$

При любом ходе Пети выигрывает Вова^{ант} $\Rightarrow$ Вова^{ант} имеет выигрышную стратегию. Вове^{ант} предстоит один ход.

$$\delta) (8; 34) \xrightarrow{\Pi} (8; 68)$$

Позиция выигрышная для Пети, он за один ход может выиграть.

$$2) \text{ а) } (6; 34) \xrightarrow{\Pi} (6; 35) \xrightarrow{B} (7; 35) \xrightarrow{\Pi} (7; 70)$$

При любом ходе Вовы, Пета выигрывает со второго хода.

$$(12; 35) \xrightarrow{\Pi} (12; 70)$$

$$(6; 70) \xrightarrow{\Pi} (7; 70)$$

$$(6; 36) \xrightarrow{\Pi} (6; 72)$$

$$\delta) (7; 34) \xrightarrow{\Pi} (7; 68) \quad (7; 68, 2) \quad 7; 70$$

$$(7; 35)$$

$$(8; 34)$$

$$(14; 34)$$

$$2) \delta) (7; 34) \xrightarrow{\Pi} (8; 34)$$

Позиция выигрышная для Пети.
2 хода.

$$\xrightarrow{B} (9; 34) \quad (8; 35) \quad (8; 68) \quad (16; 34)$$

$$\xrightarrow{\Pi} (8; 68) \quad (8; 70) \quad (8; 68) \quad (16; 68)$$

$$1) \delta) (8; 34) \xrightarrow{\Pi} (16; 34) \xrightarrow{B} (16; 68)$$

$$(9; 34) \xrightarrow{\Pi} (9; 68)$$

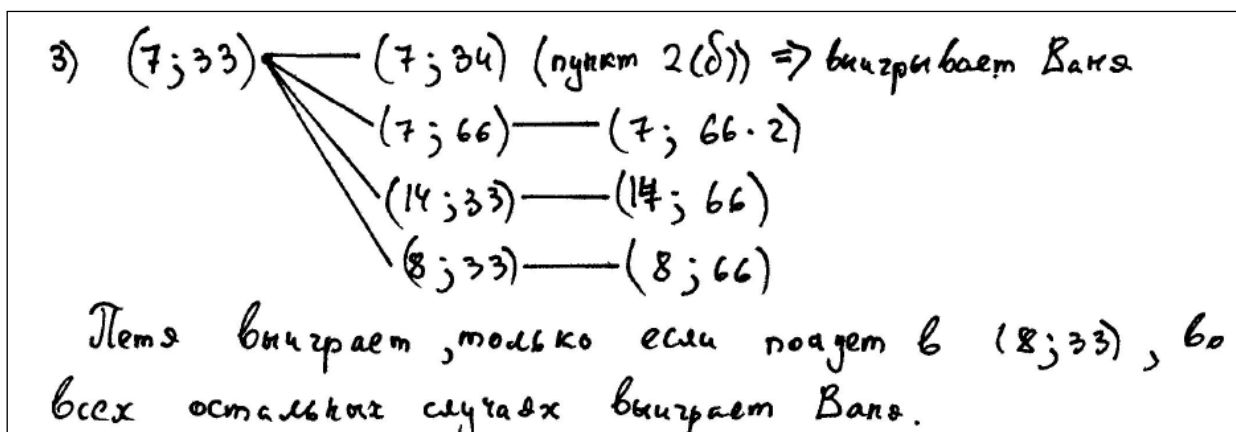
$$(8; 35) \xrightarrow{\Pi} (8; 70)$$

$$(8; 68) \xrightarrow{\Pi} (9; 68)$$

У Вовы есть стратегия, при любом ходе Пети Вова выигрывает с первого хода.

$$2) \text{ б) } (8; 33) \xrightarrow{\Pi} (8; 34) \quad (1 \delta)$$

Пета выигрывает, т.к. Вова попадает в положение $(8; 34)$, аналогично игре (1δ) .



Эталонная оценка для данного ответа: 2 (пункт 3 выполнен неверно)

Практическая работа №4

Текст задания

Оцените согласно критериям оценивания выполнение задания 27 КИМ ЕГЭ по информатике. Оценки выставьте в протокол оценивания:

В физической лаборатории проводится долговременный эксперимент по изучению гравитационного поля Земли. По каналу связи каждую минуту в лабораторию передаётся положительное целое число – текущее показание прибора «Сигма 2015». Количество передаваемых чисел в серии известно и не превышает 10 000. Все числа не превышают 1000. Временем, в течение которого происходит передача, можно пренебречь.

Необходимо вычислить «бета-значение» серии показаний прибора – минимальное **чётное** произведение двух показаний, между моментами передачи которых прошло не менее 7 минут. Если получить такое произведение не удаётся, ответ считается равным –1.

Вам предлагается два задания, связанных с этой задачей: задание А и задание Б. Вы можете решать оба задания или одно из них по своему выбору.

*Итоговая оценка выставляется как **максимальная** из оценок за задания А и Б. Если решение одного из заданий не представлено, то считается, что оценка за это задание – 0 баллов.*

Задание Б является усложнённым вариантом задания А, оно содержит дополнительные требования к программе.

А. Напишите на любом языке программирования программу для решения поставленной задачи, в которой входные данные будут запоминаться в массиве, после чего будут проверены все возможные пары элементов. Перед программой укажите версию языка программирования.

ОБЯЗАТЕЛЬНО укажите, что программа является решением ЗАДАНИЯ А.

Максимальная оценка за выполнение задания А – 2 балла.

Б. Напишите программу для решения поставленной задачи, которая будет эффективна как по времени, так и по памяти (или хотя бы по одной из этих

характеристик).

Программа считается эффективной по времени, если время работы программы пропорционально количеству полученных показаний прибора N , т.е. при увеличении N в k раз время работы программы должно увеличиваться не более чем в k раз.

Программа считается эффективной по памяти, если размер памяти, использованной в программе для хранения данных, не зависит от числа N и не превышает 1 килобайта.

Перед программой укажите версию языка программирования и кратко опишите использованный алгоритм.

ОБЯЗАТЕЛЬНО укажите, что программа является решением ЗАДАНИЯ Б.

Максимальная оценка за правильную программу, эффективную по времени и по памяти, – 4 балла.

Максимальная оценка за правильную программу, эффективную по времени, но неэффективную по памяти, – 3 балла.

НАПОМИНАЕМ! Не забудьте указать, к какому заданию относится каждая из представленных Вами программ.

Входные данные представлены следующим образом. В первой строке задаётся число N – общее количество показаний прибора. Гарантируется, что $N > 7$. В каждой из следующих N строк задаётся одно положительное целое число – очередное показание прибора.

Пример входных данных:

12
12
45
5
3
14
17
23
21
20
19
18
17

Программа должна вывести одно число – описанное в условии произведение, либо -1 , если получить такое произведение не удаётся.

Пример выходных данных для приведённого выше примера входных данных:

54

Критерии оценивания

Указания по оцениванию	Баллы
<i>Предварительные замечания.</i> 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. 2. Если ученик не указал, к какому заданию относится программа, или можно предположить, что ученик ошибся в идентификации программ, необходимо следовать приведённым ниже инструкциям. Случай 2.1. Ученик представил только одну программу. Следует рассматривать программу как решение задания Б и оценивать её по соответствующим критериям. Случай 2.2. Ученик представил две программы, но указание задания есть только для одной из программ. Следует рассматривать вторую программу как ответ на оставшееся задание. Случай 2.3. Ученик представил две программы; ни для одной из них задание не указано, или в обоих решениях указано одно и то же задание. Следует первую (по порядку в представленных учеником материалах) программу рассматривать как ответ на задание А, а вторую – как ответ на задание Б. Случай 2.4. Ученик представил более двух программ. Следует рассматривать только две последние программы и соотносить их с заданиями по правилам 2.1–2.3. Случай 2.5. Решение, представленное в качестве решения задания А, по критериям для задания Б может быть оценено в 3 или 4 балла. При этом решение, представленное в качестве решения задания Б, получило меньшую оценку. Следует считать, что ученик перепутал обозначения заданий и оценивать решение, представленное как решение задания А, по критериям задания Б. НАПОМИНАЕМ! <i>Итоговый балл за задачу – это больший</i>	

<i>из баллов, полученных учеником за каждое из двух представленных решений.</i> <i>Пояснения для проверяющих.</i> 1. Задание Б является усложнением задания А. Если в качестве решения задания Б представлено решение задания А, то согласно приведённым ниже критериям его оценка будет такой же, как если бы это решение было представлено в качестве решения задания А. 2. Два задания (и, соответственно, возможность для экзаменуемого представить две программы) дают ученику возможность (при его желании) сначала написать менее сложное и менее эффективное решение (задание А), которое даёт ему право получить 2 балла, а затем приступить к поиску более эффективного решения. 3. Приведённые в п. 2.1–2.5 правила имеют целью избежать снижения оценки из-за того, что ученик перепутал обозначения заданий	
Критерии оценивания задания А	
Программа решает поставленную задачу для любых соответствующих условию входных данных. Например, допускается переборное решение, аналогичное приведённой выше программе 4. Допускается до семи синтаксических и приравненных к ним ошибок (см. критерии оценивания задания Б на 4 балла). Допускается до двух содержательных ошибок, описанных в критериях оценивания задания Б на 3 балла	2
Не выполнены условия, позволяющие поставить 2 балла. Из описания алгоритма или общей структуры программы видно, что экзаменуемый в целом правильно представляет путь решения задачи независимо от эффективности. При этом программа может быть представлена отдельными фрагментами, без ограничений на количество синтаксических и содержательных ошибок. 1 балл ставится также за решения, верные лишь в частных случаях	1
Не выполнены критерии, позволяющие поставить 1 или 2 балла	0
<i>Максимальный балл для задания А</i>	2
Критерии оценивания задания Б	
Программа правильно работает для любых соответствующих условию входных данных и при этом эффективна как по времени, так и по памяти, т.е. не используются массивы и другие структуры данных, размер которых зависит от количества входных элементов, а время работы пропорционально этому количеству. Возможно	4

использование массивов и динамических структур данных (например, контейнеры STL в программе на языке C++) при условии, что в них в каждый момент времени хранится фиксированное количество элементов, требующих для хранения меньше 1кб (минимально необходимое количество – семь; допускается решение с запасом). Программа может содержать не более трёх синтаксических ошибок следующих видов: – пропущен или неверно указан знак пунктуации (запятая, точка с запятой, скобки и т.д.); – неверно написано или пропущено служебное слово языка программирования; – не описана или неверно описана переменная; – применяется операция, недопустимая для соответствующего типа данных. К синтаксическим ошибкам приравнивается использование неверного типа данных (например, использование целого типа вместо вещественного для представления данных при вводе и обработке). Если одна и та же ошибка встречается несколько раз, она считается за одну ошибку	
Не выполнены условия, позволяющие поставить 4 балла. Программа правильно работает для любых соответствующих условию входных данных, время работы пропорционально количеству входных элементов. Размер используемой памяти не имеет значения и может зависеть от объёма входных данных. В частности, допускается использование одного или нескольких массивов размера N (как в приведённой выше программе 3). Программа может содержать не более пяти синтаксических и приравненных к ним ошибок, описанных в критериях на 4 балла. Кроме того, допускается наличие не более одной содержательной ошибки из числа следующих: неверная инициализация при поиске минимального значения; неверная обработка начальных элементов данных, которая может, например, привести к получению ошибочного ответа при $7 < N < 14$; неточное определение границ массива, выход за границу массива (например, описан массив с границами от 1 до 7, а реально используется от 0 до 6 или наоборот); вычисленный индекс элемента массива на 1 отличается от верного; используется операция "<" вместо "<=", "or" вместо "and" и т.п.; не учитывается, что заданные показания могут начинаться	3

с одного или нескольких чётных чисел; не учитывается, что для данного набора показаний может не быть ни одного удовлетворяющего условиям произведения	
Не выполнены условия, позволяющие поставить 3 или 4 балла. Программа работает в целом верно, эффективно или нет. Например, допускается переборное решение, аналогичное приведённой выше программе 4. Допускается до семи синтаксических и приравненных к ним ошибок (см. критерии на 4 балла). Допускается до двух содержательных ошибок, описанных в критериях на 3 балла	2
Не выполнены условия, позволяющие поставить 2, 3 или 4 балла. Из описания алгоритма или общей структуры программы видно, что экзаменуемый в целом правильно представляет путь решения задачи независимо от эффективности. При этом программа может быть представлена отдельными фрагментами, без ограничений на количество синтаксических и содержательных ошибок. 1 балл ставится также за решения, верные лишь в частных случаях	1
Не выполнены критерии, позволяющие поставить 1, 2, 3 или 4 балла	0
<i>Максимальный балл для задания Б</i>	4
<i>Итоговый максимальный балл</i>	4

Образец ответа участника экзамена

Задача A

Паскаль

const

N = 10 000;

var

a: array[1..N] of integer,
count, i, j, k: integer;

~~begin~~

begin

for i := 1 to N do
 readln(a[i]);

~~for i := 1 to (N-7)~~

 count := 0

 j := 0

 k := 1.000 000

 for i := 0 to (N-7) do

 if (a[i] * a[i+7]) mod 2 = 0 then

 begin

 j := a[i] * a[i+7]

~~if~~ count := 1

 if k > j then

 k := j;

 end;

 if count = 1 then

 writeln(k)

 else

 writeln("-1")

Эталонная оценка для данного ответа: 0 (пункт3 выполнен неверно)

Практическая работа № 5

Текст задания

На основании критериев оценивания заданий с развернутым ответом варианта ЕГЭ, представленного в приложенном файле, осуществите оценивание представленных экзаменационных работ. Заполните протокол оценивания.

Критерии оценивания экзаменационных работ: полный комплект критериев оценивания к варианту (24 страницы). Все работы на зачете оцениваются по одному комплекту критериев.

Образец экзаменационной работы для оценивания (4 листа)

24.

1. Программа выведет:

2 70

2. Пример, последовательности: 3, 5, 6, 7.

3. На языке C++

функция, в которой допущена ошибка

Неправильная функция

1. `sum = x;`

2. `if (sum > 0)`

`sum = sum + x;`

`if (count > 0)`

25.

Язык программирования "C++"

~~константа~~ `K = 10001`

`for (i = 0; i < N; i++) {`

`if (a[i] < K && a[i] % 4 == 0)`

`K = a[i];`

`}`

`for (i = 0; i < N; i++) {`

~~а[а[i] % 4 == 0]~~

`if (a[i] % 4 == 0) {`

`a[i] = a[i] + K;`

`printf("%d\n", a[i]);`

`}`

26.

1. а) $22 \leq S \leq 42$, Тогда проверяем, ^{элементарная} ~~какая-то~~ наименьшее значение $S \leq 21$ не подходит, т.к. $21 \cdot 2 < 43 \Leftrightarrow 42 < 43$.

б) $S = 21$:
Ответ: $[21; 42]$

	Хочу	Тема	Банк
Нал. Намечу	$S=21$	$\xrightarrow{+1} 22$	$\xrightarrow{-2} 44 \geq 43$
		$\xrightarrow{+3} 24$	$\xrightarrow{-2} 48 \geq 43$
		$\xrightarrow{-2} 42$	$\xrightarrow{-2} 84 \geq 43$

$\Rightarrow$ Банк выиграл.

Ответ: $S=21$;

2.	$S=18, S=20$	Хочу	Тема	Банк	Тема	Банк	Тема	Банк
				$\xrightarrow{+1} 22$	$\xrightarrow{-2} 44 \geq 43$			
				$\xrightarrow{+3} 24$	$\xrightarrow{-2} 48 \geq 43$			
				$\xrightarrow{-2} 42$	$\xrightarrow{-2} 84 \geq 43$			

$\Rightarrow$ Банк выиграл.

		Хочу	Тема	Банк	Тема	Банк	Тема	Банк
		$S=20$	$\xrightarrow{+1} 21$	$\xrightarrow{+1} 22$	$\xrightarrow{-2} 44 \geq 43$			
			$\xrightarrow{+3} 24$	$\xrightarrow{-2} 48 \geq 43$				
			$\xrightarrow{-2} 42$	$\xrightarrow{-2} 84 \geq 43$				

$\Rightarrow$ Банк выиграл.

3. Ответ: $S=18, S=20$;

3.	$S=17$:	Хочу	Тема	Банк	Тема	Банк	Тема	Банк
		$S=17$	$\xrightarrow{+1} 18$	$\xrightarrow{+3} 21$	$\xrightarrow{+1} 22$	$\xrightarrow{-2} 44 \geq 43$		
			$\xrightarrow{+3} 20$	$\xrightarrow{+1} 21$	$\xrightarrow{+3} 24$	$\xrightarrow{-2} 48 \geq 43$		
			$\xrightarrow{-2} 34$	$\xrightarrow{-2} 68 \geq 43$	$\xrightarrow{-2} 42$	$\xrightarrow{-2} 84 \geq 43$		

$\Rightarrow$ Банк выиграл.

Ответ: $S=17$.

27. I Вывести программу:

Создать массив из 12000 элементов, потом найти по нему двойным циклом и вывести все пары чисел к экрану:

```
#include <stdio.h>

int main(void) {
    int N, i, a[12000], m1, m2, max = 0, s, j;
    scanf("%d", &N);
    for(i=0; i<12000; i++)
    for(i=0; i<N; i++) scanf("%d", &a[i]);
    for(i=0; i<N; i++) {
        for(j=0; j<N; j++) {
            if (i != j && a[i] < a[j]) {
                s = a[i] + a[j];
                if (s > max && s % 107 == 0 && (i < j && a[i] > a[j] || i > j && a[j] > a[i]))
                {
                    max = s;
                    m1 = a[i];
                    m2 = a[j];
                }
            }
        }
    }
    printf("%d %d", m1, m2);
    return 0;
}
```

Лист 3

II Вариант программы:

Нужно отсортировать массив на 107 элементов, попарно сравнивая соседние элементы.

```
#include <stdio.h>

int main (void) {
    int N, i, k, s, c, max=0, m1, m2;
    int a[107], b[107];
    scanf ("%d", &N);
    for(i=0; i<107; i++) a[i]=0;
    for(i=0; i<N; i++){
        scanf ("%d", &k);
        c = k % 107;
        if (a[i] > a[c]) {
            a[c] = a[i];
            b[c] = i;
        }
    }
    for(i=0; i<107; i++){
        if (a[i] > 0 && a[107-i] > 0) {
            s = a[i] + a[107-i];
            if (s > max && (a[i] > a[107-i] && b[i] < b[107-i] || a[i] < a[107-i] && b[i] > b[107-i])) {
                max = s;
                m1 = a[i];
                m2 = a[107-i];
            }
        }
    }
    printf ("%d %d", m1, m2);
}
```

Лист 4

Комплект эталонных оценок для каждой работы. (Для приведенной работы оценки 3, 2, 3, 1.)

Для подготовки практических работ используется открытый вариант ЕГЭ 2019 г. (в том числе критерии оценивания) и предоставленные РЦОИ города Москвы сканы экзаменационных работ участников экзамена.

В качестве эталонных определяются оценки, выставленные экспертами ПК в итоговом протоколе оценивания (с учетом третьей проверки и апелляций). Для разбора в ходе тренинга в том числе используются работы, сложные для оценивания. Количество работ для оценивания на каждой практической работе определяется исходя из норматива работы эксперта ПК (5 работ в час).

3.2. Итоговая аттестация включает

- итоговое тестирование на знание нормативных документов ГИА и регламента проведения экзамена;
 - зачет: оценивание заданий с развернутым ответом.
- Слушатель аттестован, если выполнил тест и на зачете суммарное расхождение с эталонными ответами не превысило 15%.

Пример тестового задания:

Отметьте документы, определяющие порядок формирования и деятельности ПК:

- 1) Положение о предметной комиссии.
- 2) Положение о ГЭК субъекта РФ.
- 3) Федеральный закон № 273-ФЗ от 29.12.2012 г. «Об образовании в Российской Федерации».

Пример кейсового задания:

1. Проконсультировать эксперта по проверке заданий с развернутым ответом во время работы ПК ГИА на пункте проверки заданий (ППЗ) может *(Выберите все верные варианты ответа):*

- 1) старший эксперт, находящийся в аудитории
- 2) организатор в аудитории
- 3) консультант, назначенный руководителем ПК
- 4) председатель ПК ГИА
- 5) руководитель ППЗ

Материалы для проведения зачета предоставляются ФГБНУ ФИПИ, РЦОИ города Москвы.

Пример зачетного задания

Для подготовки зачета используется один из вариантов ЕГЭ 2019 г. (в том числе критерии оценивания) и предоставленные РЦОИ города Москвы сканы экзаменационных работ участников экзамена. Для зачета используются экзаменационные работы, отличные от использовавшихся при выполнении практических работ. Сложность оцениваемых работ должна соответствовать средней сложности, используются работы, оцененные обоими экспертами одинаковыми баллами. В качестве эталонных определяются оценки, выставленные экспертами ПК в итоговом протоколе оценивания. Количество работ для оценивания на зачете определяется исходя из норматива работы эксперта ПК (5 работ в час).

Пример зачетного задания

Текст задания: На основании критериев оценивания заданий с развернутым ответом варианта ЕГЭ, представленного в приложенном файле, осуществите оценивание представленных экзаменационных работ. Заполните протокол оценивания.

Критерии оценивания экзаменационных работ: полный комплект

критериев оценивания к варианту (24 страницы). Все работы на зачете оцениваются по одному комплекту критериев.

Образец экзаменационной работы для оценивания (4 листа)

24.

1. Программа выведет:

2 70

2. Пример, последовательности: 3, 5, 6, 7.

3. На языке C++

функция, в которой функция ошибки

Управляемая функция

1. $sum = x;$

2. $if (sum > 0)$

$sum = sum + x;$

$if (count > 0)$

25.

Язык программирования

"C++":

~~const~~ $k = 10001$

$for (i = 0; i < N; i++) \{$

$if (a[i] < k \ \&\& \ a[i] \% 4 == 0)$

$k = a[i];$

$\}$

$for (i = 0; i < N; i++) \{$

~~if (a[i] % 4 == 0)~~

$if (a[i] \% 4 == 0) \{$

$a[i] = a[i] + k;$

$printf ("%d\n", a[i]);$

$\}$

Лист 1

26.

1. а) $22 \leq S \leq 42$, Тогда проверяем, ^{элементарная} ~~какая-то~~ наименьшее значение $S \leq 21$ не подходит, т.к. $21 \cdot 2 < 43 \Leftrightarrow 42 < 43$.

б) $S = 21$:
Ответ: $[21; 42]$

	Хочу	Тема	Банк
Нал. Намечу	$S=21$	$\xrightarrow{+1} 22$	$\xrightarrow{-2} 44 \geq 43$
		$\xrightarrow{+3} 24$	$\xrightarrow{-2} 48 \geq 43$
		$\xrightarrow{-2} 42$	$\xrightarrow{-2} 84 \geq 43$

$\Rightarrow$ Банк выиграл.

Ответ: $S=21$;

2.	$S=18, S=20$	Хочу	Тема	Банк	Тема	Банк	Тема	Банк
		$S=18$	$\xrightarrow{+3} 21$	$\xrightarrow{+1} 22$	$\xrightarrow{-2} 44 \geq 43$			
			$\xrightarrow{-2} 42$	$\xrightarrow{+3} 24$	$\xrightarrow{-2} 48 \geq 43$			
				$\xrightarrow{-2} 42$	$\xrightarrow{-2} 84 \geq 43$			

$\Rightarrow$ Банк выиграл.

	Хочу	Тема	Банк	Тема	Банк
Тема	$S=20$	$\xrightarrow{+1} 21$	$\xrightarrow{+1} 22$	$\xrightarrow{-2} 44 \geq 43$	
		$\xrightarrow{+3} 24$	$\xrightarrow{-2} 48 \geq 43$		
		$\xrightarrow{-2} 42$	$\xrightarrow{-2} 84 \geq 43$		

$\Rightarrow$ Банк выиграл.

3. Ответ: $S=18, S=20$;

3.	$S=17$:	Хочу	Тема	Банк	Тема	Банк	Тема	Банк
		$S=17$	$\xrightarrow{+1} 18$	$\xrightarrow{+3} 21$	$\xrightarrow{+1} 22$	$\xrightarrow{-2} 44 \geq 43$		
			$\xrightarrow{+3} 20$	$\xrightarrow{+1} 21$	$\xrightarrow{+3} 24$	$\xrightarrow{-2} 48 \geq 43$		
			$\xrightarrow{-2} 34$	$\xrightarrow{-2} 68 \geq 43$	$\xrightarrow{-2} 42$	$\xrightarrow{-2} 84 \geq 43$		

$\Rightarrow$ Банк выиграл.

Ответ: $S=17$.

27.

I Вывести программу:

Создать массив из 12000 элементов, потом из них по двум двойным циклам и вывести все пары чисел к экрану:

```
#include <stdio.h>

int main(void) {
    int N, i, a[12000], m1, m2, max = 0, s, j;
    scanf("%d", &N);
    for(i=0; i<12000; i++)
    for(i=0; i<N; i++) scanf("%d", &a[i]);
    for(i=0; i<N; i++) {
        for(j=0; j<N; j++) {
            if (i != j && a[i] < a[j]) {
                s = a[i] + a[j];
                if (s > max && s % 107 == 0 && (i < j && a[i] > a[j] || i > j && a[j] > a[i]))
                {
                    max = s;
                    m1 = a[i];
                    m2 = a[j];
                }
            }
        }
    }
    printf("%d %d", m1, m2);
    return 0;
}
```

Лист 3

II Вариант программы:

Нужно отсортировать элементы на 107-й каждого элемента, попарно считывая соответствующие значения элементов.

```
#include <stdio.h>

int main (void) {
    int N, i, k, s, c, max=0, m1, m2;
    int a[107], b[107];
    scanf ("%d", &N);
    for(i=0; i<107; i++) a[i]=0,
    for(i=0; i<N; i++){
        scanf ("%d", &k);
        c = k % 107;
        if (a[c] > a[i]) {
            a[c] = a[i];
            b[c] = i;
        }
    }
    for(i=0; i<107; i++){
        if (a[i] > 0 && a[107-i] > 0) {
            s = a[i] + a[107-i];
            if (s > max || (a[i] > a[107-i] && b[i] < b[107-i] || a[i] < a[107-i] && b[i] > b[107-i])) {
                max = s;
                m1 = a[i];
                m2 = a[107-i];
            }
        }
    }
    printf ("%d %d", m1, m2);
}
```

Лист 4

Комплект эталонных оценок для каждой работы. (Для приведенной работы оценки 3, 2, 3, 1.)

Раздел 4. «Организационно-педагогические условия реализации программы»

4.1. Учебно-методическое обеспечение и информационное обеспечение программы

Нормативные документы ГИА-11

1. Федеральный закон от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации» (в действующей редакции)
2. Постановление Правительства Российской Федерации от 31.08.2013 № 755 «О федеральной информационной системе обеспечения проведения государственной итоговой аттестации обучающихся, освоивших основные образовательные программы основного общего и среднего общего образования, и приема граждан в образовательные организации для получения среднего профессионального и высшего образования и региональных информационных системах обеспечения проведения государственной итоговой аттестации обучающихся, освоивших основные образовательные программы основного общего и среднего общего образования» (в действующей редакции)
3. Приказ Федеральной службы по надзору в сфере образования и науки от 18.06.2018 № 831 «Об утверждении требований к составу и формату сведений, вносимых и передаваемых в процессе репликации в федеральную информационную систему обеспечения проведения государственной итоговой аттестации обучающихся, освоивших основные образовательные программы основного общего и среднего общего образования, и приема граждан в образовательные организации для получения среднего профессионального и высшего образования и региональные информационные системы обеспечения проведения государственной итоговой аттестации обучающихся, освоивших основные образовательные программы основного общего и среднего общего образования, а также к срокам внесения и передачи в процессе репликации сведений в указанные информационные системы».
4. Приказ Министерства просвещения Российской Федерации № 190, Федеральной службы по надзору в сфере образования и науки № 1512 от 07.11.2018 «Об утверждении Порядка проведения государственной итоговой аттестации по образовательным программам среднего общего образования».
5. Приказ Федеральной службы по надзору в сфере образования и науки № 1274 от 17.12.2013 «Об утверждении Порядка разработки, использования и хранения контрольных измерительных материалов при проведении государственной итоговой аттестации по образовательным программам основного общего образования и порядка разработки, использования и хранения контрольных измерительных материалов при проведении государственной итоговой аттестации по образовательным программам среднего общего образования» (в действующей редакции).
6. Приказ Федеральной службы по надзору в сфере образования и науки от 26 июня 2019 г. № 876 «Об определении минимального количества баллов

единого государственного экзамена, подтверждающего освоение образовательной программы среднего общего образования, и минимального количества баллов единого государственного экзамена, необходимого для поступления в образовательные организации высшего образования на обучение по программам бакалавриата и программам специалитета».

7. Постановление Правительства Москвы №184-ПП от 04.05.2011 «Об утверждении Порядка выплаты и определения размера компенсации работникам, привлекаемым к проведению единого государственного экзамена, за работу по его подготовке и проведению» (в действующей редакции).

8. Приказ Департамента образования города Москвы от 18.05.2018 г. № 190 «О выплате компенсации педагогическим работникам и лицам, осуществляющим обработку экзаменационных работ, участвующим в подготовке и проведении государственной итоговой аттестации по образовательным программам основного общего и среднего общего образования» (в действующей редакции).

9. Приказ Департамента образования города Москвы от 16.10.2018 № 408 «О распределении функций по организации и проведению государственной итоговой аттестации по образовательным программам основного общего и среднего общего образования в 2019 году».

10. Приказ Департамента образования города Москвы от 01.02.2019 г. № 32 «Об утверждении состава комиссии по присвоению статуса эксперта предметных комиссий при проведении государственной итоговой аттестации по образовательным программам среднего общего образования и формы сертификата».

11. Приказ Департамента образования города Москвы от 01.02.2019 г. № 30 «Об утверждении Порядка формирования предметных комиссий при проведении государственной итоговой аттестации по образовательным программам среднего общего образования».

12. Приказ Департамента образования города Москвы от 29.12.2018 г. № 514 «Об утверждении Положения о предметных комиссиях при проведении государственной итоговой аттестации по образовательным программам среднего общего образования».

13. Приказ Департамента образования города Москвы от 28.12.2015 №3586 «Об утверждении Положения о конфликтной комиссии при проведении государственной итоговой аттестации по образовательным программам среднего общего образования» (в действующей редакции).

14. Документы, определяющие структуру и содержание контрольных измерительных материалов единого государственного экзамена в 2020 г.

15. Иные федеральные, ведомственные и региональные нормативные акты, методические материалы по организации и проведению государственной итоговой аттестации по образовательным программам среднего общего образования (обновляются ежегодно).

Основная литература

1. Спецификация контрольных измерительных материалов для проведения в 2020 году единого государственного экзамена по информатике. – М.: ФИПИ, 2019.
2. Демонстрационный вариант единого государственного экзамена по информатике в 2020 году. – М.: ФИПИ, 2019.
3. Кодификатор элементов содержания и требований к уровню подготовки участников экзамена общеобразовательных учреждений для проведения единого государственного экзамена по информатике в 2020 году. – М.: ФИПИ, 2019.

Интернет-ресурсы

1. Демоверсии, спецификации, кодификаторы. ФИПИ. – <http://fipi.ru/ege-i-gve-11/demoversii-specifikacii-kodifikatory>.
2. Для предметных комиссий субъектов РФ. ФИПИ. – <http://fipi.ru/ege-i-gve-11/dlya-predmetnyh-komissiy-subektov-rf>.
3. Открытый банк заданий. ФИПИ. - <http://fipi.ru/content/otkrytyy-bank-zadaniy-ege>

4.2. Материально-технические условия реализации программы

Аудитории, оборудованные аудио-, видеоаппаратурой, с доступом к сети Интернет.