

DOI: 10.32517/2221-1993-2025-24-1-77-84



Т. В. Шамаева

Школа № 2007 ФМШ, г. Москва, Россия

ИСПОЛЬЗОВАНИЕ ЭЛЕКТРОННЫХ ТАБЛИЦ В РЕШЕНИИ ЗАДАНИЙ КЕГЭ ПО ИНФОРМАТИКЕ*

Аннотация

В статье предлагаются методы решения некоторых заданий КЕГЭ по информатике с помощью электронных таблиц как альтернатива традиционно принятому способу решения задач с помощью написания программ. В частности, рассматриваются задания 13, 23, 26, представленные в демоверсии 2025 года.

В силу того что на изучение темы «Программирование» не всегда есть возможность отвести достаточное количество часов в рамках школьного курса, а также из-за того что не все учащиеся успевают глубоко овладеть навыками написания кода, вариант решения заданий с помощью табличных процессоров выглядит весьма реалистичным и привлекательным. Нужно отметить, что во многих случаях таким способом удается решать задания достаточно быстро и точно, так как электронные таблицы позволяют наглядно представить обрабатываемые данные и обучающимся легче в них сориентироваться и найти требуемый ответ.

Статья может быть полезна школьникам, планирующим сдавать КЕГЭ, преподавателям информатики для подготовки учеников к экзамену, а также для изучения темы «Электронные таблицы» в школьном курсе информатики.

Решения задач в статье выполнены с помощью электронного процессора LibreOffice Calc. К статье прилагаются дополнительные материалы в виде файлов с решениями в формате указанного процессора.

Ключевые слова: КЕГЭ, КИМ, информатика, демоверсия, электронная таблица, IP-адрес, маска, сортировка.

Информация об авторе

Шамаева Татьяна Владимировна, учитель информатики, школа № 2007 ФМШ, г. Москва, Россия; *e-mail*: shamaeva-tv@fms2007.ru

1. Введение

Требования к базовому и углубленному уровням изучения информатики, зафиксированные в ФГОС среднего общего образования [16], включают в себя раздел «Информационные технологии», который охва-

тывает вопросы применения информационных технологий, реализованных в прикладных программных продуктах и интернет-сервисах, использования баз данных и электронных таблиц для решения прикладных задач.

Электронные таблицы — мощный компьютерный инструмент, широко применяемый для автоматизации

* Материалы к статье можно скачать на сайте ИНФО: http://infojournal.ru/journals/school/school_01-2025/

T. V. Shamaeva

School 2007 FMSH, Moscow, Russia

THE USE OF SPREADSHEETS IN SOLVING TASKS FROM THE COMPUTER BASED UNIFIED STATE EXAM IN INFORMATICS

Abstract

The article proposes methods for solving some tasks from the computer based Unified State Exam in informatics using spreadsheets as an alternative to the traditionally accepted method of solving problems by coding. In particular, the tasks 13, 23, 26 presented in the 2025 demo version are considered.

Due to the fact that it is not always possible to allocate a sufficient number of hours to study the theme "Programming" within the school course, and also due to the fact that not all students have time to deeply master the skills of coding, the option of solving tasks using spreadsheet processors looks very realistic and attractive. It should be noted that in many cases this method allows you to solve tasks quite quickly and accurately, since spreadsheets allow you to visually present the data being processed and it is easier for students to navigate them and find the required answer.

The article may be useful for students planning to take computer based Unified State Exam in informatics, informatics teachers to prepare students for the exam, as well as for studying the theme "Spreadsheets" in a school informatics course.

The solutions to the problems in the article are implemented using the LibreOffice Calc electronic processor. Additional materials are attached to the article in the form of files with tasks solutions in the format of the specified spreadsheets.

Keywords: computer based Unified State Exam, computer based USE, test materials, informatics, demo, spreadsheet, IP address, mask, sorting.

Information about the author

Tatyana V. Shamaeva, informatics teacher, school 2007 FMSH, Moscow, Russia; *e-mail*: shamaeva-tv@fms2007.ru

вычислений, анализа и визуализации числовых данных, математического моделирования [2]. Этой теме уделено внимание как в основной [1], так и в старшей [2] школе.

Решения некоторых заданий КЕГЭ с помощью электронных таблиц MS Excel неоднократно рассматривались различными авторами (см., например, задание 8 в работе [6], задание 22 в работе [7]).

В данной статье будут предложены решения заданий 13, 23 и 26, выполненные с помощью офисного пакета LibreOffice (версия 7.3.3.2 (x64)). Для знакомства с программами LibreOffice предусмотрена возможность использования документации [19] и различных учебных материалов (см., например, [2, 14]).

Согласно кодификатору ([4], Кодификатор, п. 2.13 «Проверяемые требования к предметным результатам освоения основной образовательной программы»), в КЕГЭ 2025 проверяется «умение использовать электронные таблицы для анализа, представления и обработки данных». Этот навык и будет использоваться для решения задач, представленных в данной статье.

2. Задание 13

В задании проверяется умение «использовать маску подсети и разделять IP-сети на подсети с помощью масок подсетей» ([4], Кодификатор, п. 1.2). Теория, необходимая для понимания данной задачи, изложена в учебнике К. Ю. Полякова, Е. А. Еремина [12]. В статье К. Ю. Полякова [11] предложен способ решения этой задачи с помощью написания программ на языках программирования Python и PascalABC.Net с подключением специальных библиотек. В данной статье мы рассмотрим вариант решения без программирования, только лишь с помощью встроенных функций табличного процессора. В статье А. А. Емельяновой [5] для получения результата также применяется табличный процессор, но решение основано на переборе вариантов и удобно в использовании для сетей только с небольшим количеством узлов.

Задача Демо-25.

Сеть задана IP-адресом 172.16.168.0 и маской сети 255.255.248.0. Сколько в этой сети IP-адресов, для ко-

торых количество единиц в двоичной записи IP-адреса не кратно 5?

Решение.

Решим задачу, используя правила комбинаторики и электронные таблицы LibreOffice Calc.

Переведем значения октетов в маску и сети в двоичный код с помощью функции ДЕС.В.ДВ() и запишем ответ в восьми разрядах (рис. 1). Формулу удобно распространить из ячейки A1 с помощью маркера автозаполнения.

Маска сети определяет, какая часть IP-адреса узла относится к адресу сети, а какая — к адресу самого узла в этой сети. Значения третьего октета маски покажут, где заканчиваются разряды, соответствующие адресу сети (под «1» маски). Разряды адреса узла под нулевыми разрядами маски относятся к адресу компьютера внутри сети и могут варьироваться (рис. 2).

В изменяемую часть можем добавить такое количество единиц, чтобы общее количество единиц не делилось на 5: $((8 + x) \% 5 \neq 0)$.

По формуле комбинаторики вычислим количество способов, которыми можно расположить единицы по 11 разрядам, т. е. вычислим количество сочетаний из 11 по добавленному числу единиц. В электронных таблицах используем функцию ЧКОМБ().

Общий вид решения в формате отображения значений и формул показан на рисунке 3.

3. Задание 23

Согласно спецификации [4, Спецификация], в задании 23 необходимо «умение анализировать ход исполнения алгоритма». Для достижения этого понадобится использовать прием «динамического программирования, как метод решения задач с сохранением промежуточных результатов, подсчет количества вариантов» [4, Кодификатор, п. 3.16]. Принципы динамического программирования, а также примеры и методы их реализации с помощью программ рассмотрены в [9, 13]. В статье А. А. Чапурных [17] также предложено программное решение этого задания на языках программирования

	A	B	C	D
1	255	255	248	0 Маска
2	172	16	168	0 Сеть
3				
4	=ДЕС.В.ДВ(A1;8)	=ДЕС.В.ДВ(B1;8)	=ДЕС.В.ДВ(C1;8)	=ДЕС.В.ДВ(D1;8)
5	=ДЕС.В.ДВ(A2;8)	=ДЕС.В.ДВ(B2;8)	=ДЕС.В.ДВ(C2;8)	=ДЕС.В.ДВ(D2;8)

Рис. 1. Перевод в двоичную систему

	A	B	C	D	E
1	255	255	248	0 Маска	
2	172	16	168	0 Сеть	
3					
4	11111111	11111111	11111000	00000000	
5	10101100	00010000	10101000	00000000	
6	8 единиц есть		11 разрядов варьируем		
7					
8					

Рис. 2. Анализ маски и адреса сети

Добавим х единиц, чтобы х + 8 не делилось на 5	Сочетания из 11 по х	7	Добавим х единиц, чтобы х + 8 не делилось на 5	Сочетания из 11 по х
↓ х	↓ C(11,х)	8	↓ х	↓ C(11,х)
0	1	12	0	=ЧКОМБ(11;В12)
1	11	13	1	=ЧКОМБ(11;В13)
3	165	14	3	=ЧКОМБ(11;В14)
4	330	15	4	=ЧКОМБ(11;В15)
5	462	16	5	=ЧКОМБ(11;В16)
6	462	17	6	=ЧКОМБ(11;В17)
8	165	18	8	=ЧКОМБ(11;В18)
9	55	19	9	=ЧКОМБ(11;В19)
10	11	20	10	=ЧКОМБ(11;В20)
11	1	21	11	=ЧКОМБ(11;В21)
Ответ:	1663	22	Ответ:	=СУММ(С12:С21)

Рис. 3. Решение задачи

Python и PascalABC.Net с помощью рекурсии. В статье В. С. Попова, Е. А. Алефиренко, Л. Ю. Черницыной [15] анализ решения с помощью программ детализируется моделированием с помощью дерева рекурсивных вызовов, уделяется также внимание меморизации как ключевому отличию метода динамического программирования от рекурсии. В данной статье, опираясь на работу [18], мы рассмотрим решение с использованием электронных таблиц. Промежуточные результаты для подсчета количества вариантов, необходимые при «динамическом» подходе, удобно сохранять в последовательных ячейках таблицы. Покажем это на примере двух задач из демо-версий последних лет [4, Демонстрационный вариант].

Задача Демо-24.

Исполнитель преобразует число на экране.

У исполнителя есть три команды, которые обозначены латинскими буквами:

- А. Прибавить 1
- В. Умножить на 2
- С. Возвести в квадрат

Программа для исполнителя — это последовательность команд. Сколько существует программ, для которых при исходном числе 2 результатом является число 20, при этом траектория вычислений не содержит числа 11?

Решение.

Используя принцип динамического программирования, будем последовательно вычислять результат (т. е. находить количество программ) для всех величин N , от начальной до конечной. Обозначим это количество как K_N . Тогда математические утверждения выглядят так:

$K_N = K_{N-1}$ — для N , не кратных 2 и не являющихся квадратом числа;

$K_N = K_{N-1} + K_{N/2}$ — для N , кратных 2 и не являющихся квадратом числа;

$K_N = K_{N-1} + K_{N^{*}0.5}$ — для N , не кратных 2 и являющихся квадратом числа*;

$K_N = K_{N-1} + K_{N/2} + K_{N^{*}0.5}$ — для N , кратных 2 и являющихся квадратом числа.

В результате задача сводится к поиску K_{20} .

На первом этапе создадим таблицу с пятью столбцами.

В первом укажем величины N от 2 до 20, в трех следующих будем хранить количество программ по каждой из операций, в последнем столбце будем хранить искомое количество программ K_N для каждого N .

Как видно из математической модели, K_{N-1} входит как слагаемое во всех случаях, поэтому удобно для второго столбца указать ссылку на общий результат предыдущей строки и распространить формулу.

Для заполнения третьего и четвертого столбцов потребуются вручную указать ссылки на ячейки или написать более сложную формулу (например, через ДВССЫЛ() или ВПР(), здесь рассматривать такие варианты не будем).

В результирующем столбце хранится сумма по всей строке, т. е. по всем трем видам операций.

Среди видов динамического программирования выделяют метод «динамика вперед». При таком подходе рассматривается, на какую позицию повлияет данная точка. При операции возведения в степень K_N войдет как слагаемое в сумму для $K_{N^{*}2}$. Поэтому для столбца, содержащего операцию возведения в степень, поставим ссылки на ячейки, из которых вычисляется квадрат (стрелки на рисунке 4). Например, для $N=4$ это K_2 , для $N=9$ это K_3 . Аналогично для второго столбца K_N войдет как слагаемое в сумму для K_{2*N} , поэтому необходимо поставить ссылки на ячейки, откуда можем получить каждую из четных величин (рис. 4).

Необходимо принять во внимание, что траектория вычислений не содержит числа 11. Чтобы учесть это требование, достаточно выставить ноль в итоговом столбце напротив $K=11$.

Задача Демо-25.

У исполнителя есть две команды, которые обозначены латинскими буквами:

- А. Вычти 2
- В. Найди целую часть от деления на 2

Программа для исполнителя — это последовательность команд. Сколько существует программ, для которых при исходном числе 38 результатом является число 2 и при этом траектория вычислений содержит число 16?

Решение.

Для решения заданий, содержащих обязательное посещение некоторой точки, удобно разбить задачу на две

* $N^{*}0.5$ обозначена операция взятия квадратного корня.

	A	B	C	D	E		A	B	C	D	E
1	N	+1	*2	**2	Общ кол K	1	N	+1	*2	**2	Общ кол K
2	2				1	2	2				1
3	3	1			1	3	3	=E2			=СУММ(B3:D3)
4	4	1	1		1	4	4	=E3	1	=E2	=СУММ(B4:D4)
5	5	3			3	5	5	=E4			=СУММ(B5:D5)
6	6	3	1		4	6	6	=E5	=E3		=СУММ(B6:D6)
7	7	4			4	7	7	=E6			=СУММ(B7:D7)
8	8	4	3		7	8	8	=E7	=E4		=СУММ(B8:D8)
9	9	7			8	9	9	=E8	=E3		=СУММ(B9:D9)
10	10	8	3		11	10	10	=E9	=E5		=СУММ(B10:D10)
11	11	11			0	11	11	=E10			0
12	12	0	4		4	12	12	=E11	=E6		=СУММ(B12:D12)
13	13	4			4	13	13	=E12			=СУММ(B13:D13)
14	14	4	4		8	14	14	=E13	=E7		=СУММ(B14:D14)
15	15	8			8	15	15	=E14			=СУММ(B15:D15)
16	16	8	7		18	16	16	=E15	=E8	=E4	=СУММ(B16:D16)
17	17	18			18	17	17	=E16			=СУММ(B17:D17)
18	18	18	8		26	18	18	=E17	=E9		=СУММ(B18:D18)
19	19	26			26	19	19	=E18			=СУММ(B19:D19)
20	20	26	11		37	20	20	=E19	=E10		=СУММ(B20:D20)

Рис. 4. Использование метода динамического программирования

подзадачи. Посчитаем количество решений от исходной точки до промежуточной, затем посчитаем количество решений от промежуточной точки до конечной. Результатом будет произведение найденных величин.

Из преимуществ такого подхода следует отметить следующие:

- этот метод гарантированно проведет траекторию через обязательную для посещения точку;
- величины, с которыми работаем, будут меньше, чем при решении задачи без разбиения, что сделает решение более наглядным и облегчит поиск ошибок;
- для задач, содержащих только операцию сложения, таблицы будут содержать одинаковые значения в итоговом столбце;
- подзадачи легко копировать с некоторой корректировкой.

Для данной задачи получим:

$$K_{38-2} = K_{38-16} * K_{16-2}.$$

Особенность этой задачи также состоит в использовании целочисленного деления на два. Нужно отметить, что один и тот же результат мы будем получать из двух точек. Например, $7 // 2 = 3$ и $6 // 2 = 3$. Это значит, что в столбце, соответствующем данной операции, результат будет суммой двух величин: $K_3 = K_7 + K_6$ (рис. 5, 6).

4. Задание 26

Согласно спецификации [4, Спецификация], в задании 26 требуется «умение обрабатывать целочисленную информацию с использованием сортировки». Это задание относится к заданиям высокого уровня сложности и предполагает поиск решения в течение 35 минут. Электронные

	B	C	D	E	F	G	H	I
1	-2	//2	K		N	-2	//2	K
2				1		16		1
3				0		15		0
4	1			1		14	1	1
5	0			0		13	0	0
6	1			1		12	1	1
7	0			0		11	0	0
8	1			1		10	1	1
9	0			0		9	0	0
10	1			1		8	1	2
11	0			0		7	0	1
12	1			1		6	2	3
13	0			0		5	1	2
14	1			1		4	3	5
15	0			0		3	2	6
16	1			1		2	5	12
17	0			0				
18	1			1				
19	0			0				
20	1			1				
21	0	1		1				
22	1	1		2				
23	1	1		2				
24	2	1		3				36

Рис. 5. Решение задачи методом динамического программирования

	A	B	C	D	E	F	G	H	I
1	N	-2	//2	K		N	-2	//2	K
2	38				1	16			1
3	37			=СУММ(B3:C3)		15			=СУММ(G3:H3)
4	36	=D2		=СУММ(B4:C4)		14	=I2		=СУММ(G4:H4)
5	35	=D3		=СУММ(B5:C5)		13	=I3		=СУММ(G5:H5)
6	34	=D4		=СУММ(B6:C6)		12	=I4		=СУММ(G6:H6)
7	33	=D5		=СУММ(B7:C7)		11	=I5		=СУММ(G7:H7)
8	32	=D6		=СУММ(B8:C8)		10	=I6		=СУММ(G8:H8)
9	31	=D7		=СУММ(B9:C9)		9	=I7		=СУММ(G9:H9)
10	30	=D8		=СУММ(B10:C10)		8	=I8	=I2	=СУММ(G10:H10)
11	29	=D9		=СУММ(B11:C11)		7	=I9	=I3+I4	=СУММ(G11:H11)
12	28	=D10		=СУММ(B12:C12)		6	=I10	=I5+I6	=СУММ(G12:H12)
13	27	=D11		=СУММ(B13:C13)		5	=I11	=I7+I8	=СУММ(G13:H13)
14	26	=D12		=СУММ(B14:C14)		4	=I12	=I9+I10	=СУММ(G14:H14)
15	25	=D13		=СУММ(B15:C15)		3	=I13	=I11+I12	=СУММ(G15:H15)
16	24	=D14		=СУММ(B16:C16)		2	=I14	=I13+I14	=СУММ(G16:H16)
17	23	=D15		=СУММ(B17:C17)					
18	22	=D16		=СУММ(B18:C18)					
19	21	=D17		=СУММ(B19:C19)					
20	20	=D18		=СУММ(B20:C20)					
21	19	=D19	=D2	=СУММ(B21:C21)					
22	18	=D20	=D3+D4	=СУММ(B22:C22)					
23	17	=D21	=D5+D6	=СУММ(B23:C23)					
24	16	=D22	=D7+D8	=СУММ(B24:C24)					=I16*D24

Рис. 6. Решение задачи методом динамического программирования, отображение формул

таблицы удобно использовать в данном задании, если алгоритм решения не подразумевает вложенных циклов, например, как в демоверсиях 2024 и 2025 годов. Такие задания можно решить значительно быстрее указанного времени с помощью табличного процессора.

Задача Демо-25.

Во время сессии студенты сдают 4 экзамена, за каждый из которых можно получить от 2 до 5 баллов. Студенты, получившие хотя бы одну «двойку», считаются не сдавшими сессию. Результаты сессии публикуются в виде рейтингового списка, в котором сначала указаны идентификационные номера студентов (ID), сдавших сессию, в порядке убывания среднего балла за сессию, а в случае равенства средних баллов — в порядке возрастания ID. Затем располагаются ID студентов, не сдавших сессию: сначала — получивших одну «двойку», затем — получивших две «двойки», потом ID студентов с тремя «двойками» и, наконец, ID студентов, получивших по 2 балла за каждый из экзаменов. Если студенты имеют одинаковое количество «двоек», то их ID в рейтинге располагаются в порядке возрастания. Повышенную стипендию получают студенты, занявшие в рейтинговом списке первые 25 % мест, при условии отсутствия у них «двоек». Гарантируется, что без «двоек» сессию сдали не менее 25 % студентов. Найдите ID студента, который занимает последнее место среди студентов с повышенной стипендией, а также ID первого в рейтинговом списке студента, который имеет более двух «двоек». В ответе запишите два целых положительных числа: сначала ID студента, который занимает последнее место среди студентов с повышенной стипендией, затем ID первого в рейтинговом списке студента, который имеет более двух «двоек».

Входные данные.

В первой строке входного файла находится число N , обозначающее количество студентов (целое положительное число, не превышающее 10 000). Каждая из следую-

щих N строк содержит 5 чисел через пробел: ID студента (целое положительное число, не превышающее 100 000) и четыре оценки, полученные им за сессию. Гарантируется, что общее число студентов N кратно 4 и хотя бы один студент имеет более двух «двоек». Во входном файле все ID различны.

Выходные данные.

Два натуральных числа: искомые ID студентов в порядке, указанном в условии задачи.

Типовой пример организации данных во входном файле:

```
8
4 4 4 4 4
7 5 5 5 2
10 3 4 4 5
1 4 4 4 3
6 3 5 5 3
2 2 2 2 2
13 2 2 2 3
3 3 3 3 3
```

При таких исходных данных рейтинговый список ID имеет вид:

```
4 6 10 1 3 7 13 2.
```

Ответ: 6 13.

Решение.

Откроем текстовый файл в LibreOffice Calc и выберем пробел в качестве разделителя (рис. 7).

Количество студентов N , находящееся в первой строке, перенесем на вторую строку любого свободного столбца и удалим первую строку. Далее в столбце F вычислим значение среднего балла для каждого студента. Для этого в ячейку F1 поместим формулу: =СРЗНАЧ(B1:E1) и распространим ее на весь столбец F. В ячейку G1 поместим формулу: =СЧЁТЕСЛИ(B1:E1;2), подсчитав количество «двоек» у студента, и также распространим эту формулу на весь столбец G.

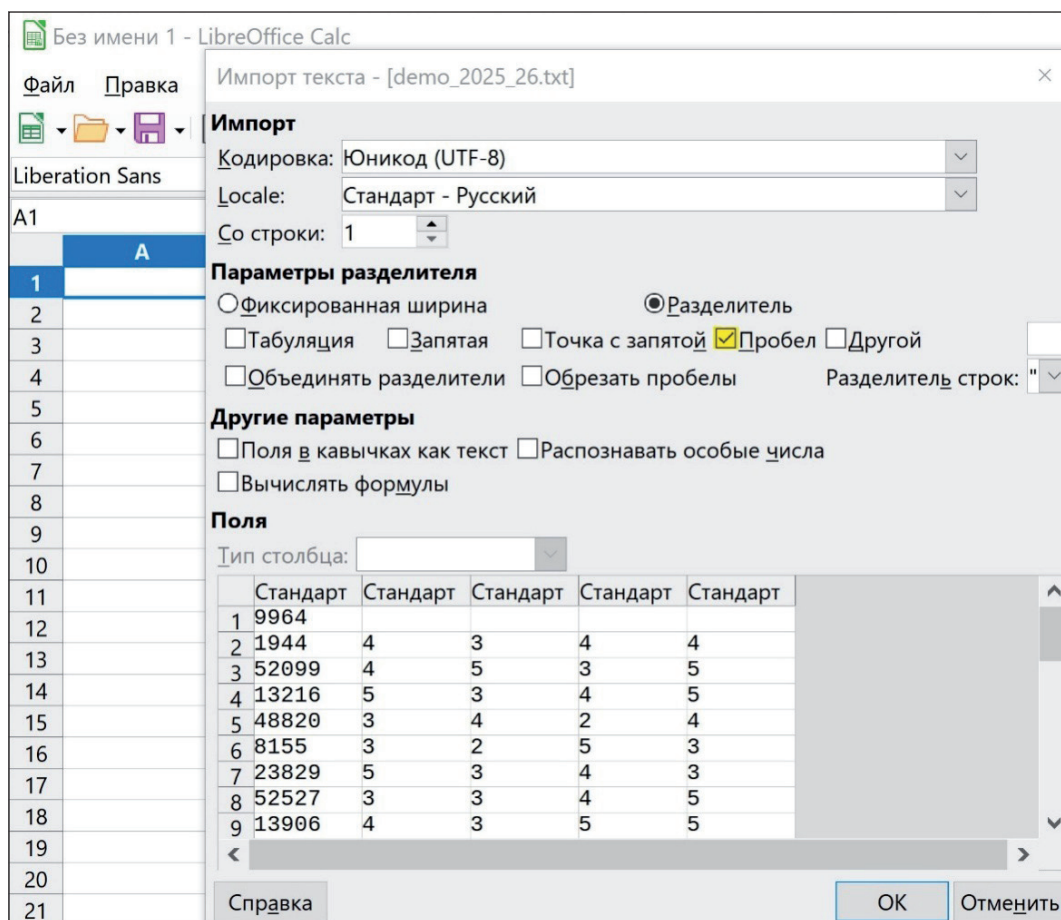


Рис. 7. Настройка параметров открытия текстового файла

Поскольку для решения задачи потребуются сортировки по нескольким параметрам, выделим все столбцы и выберем в меню *Данные, Сортировка*. В открывшемся меню используем три ключа сортировки. Прежде всего отсортируем данные по возрастанию значений столбца G, содержащего количество «двоек».

Таким образом, студенты, не имеющие «двоек», будут находиться в списке сверху. Следующий ключ (столбец F) расположит студентов с равным количеством «двоек» по убыванию среднего балла. Последний ключ применит сортировку по ID в столбце A при равенстве количества «двоек» и среднего балла (рис. 8).

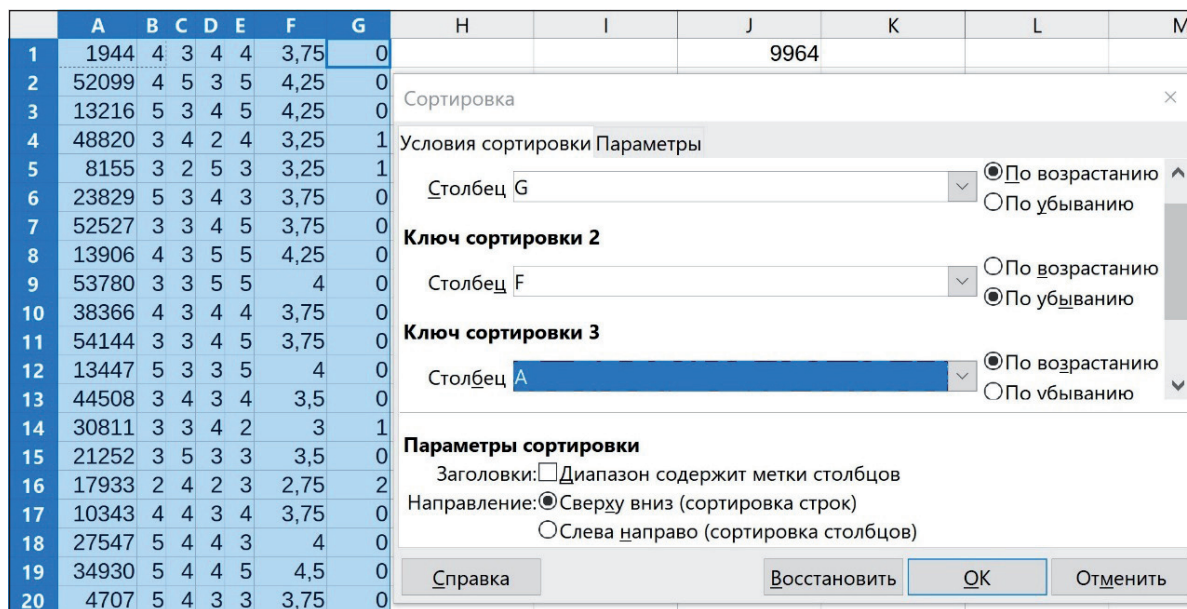


Рис. 8. Настройка параметров сортировки

	A	B	C	D	E	F	G
2487	52128	5	5	4	3	4,25	0
2488	52155	5	3	5	4	4,25	0
2489	52181	5	5	4	3	4,25	0
2490	52268	5	3	5	4	4,25	0
2491	52326	4	5	3	5	4,25	0
2492	52380	5	3	4	5	4,25	0
2493	52435	5	3	4	5	4,25	0
2494	52525	5	3	4	5	4,25	0
2495	52541	3	4	5	5	4,25	0
2496	52563	3	5	5	4	4,25	0

Рис. 9. Ответ на первый вопрос

	A	B	C	D	E	F	G
1	635	2	2	5	2	2,75	3
2	907	2	2	2	4	2,5	3
3	917	2	2	2	5	2,75	3
4	1302	2	3	2	2	2,25	3
5	1474	2	2	4	2	2,5	3
6	1532	2	4	2	2	2,5	3
7	1598	2	2	2	3	2,25	3
8	1853	2	2	2	4	2,5	3
9	2538	2	2	2	3	2,25	3
10	2708	2	2	5	2	2,75	3

Рис. 10. Ответ на второй вопрос

По отсортированным данным найдем ID студента, находящегося на строке с номером 2491 (9964/4). Таким образом, студент с ID = 52326 и есть последний в списке, кто получил повышенную стипендию. Это ответ на первый вопрос задачи (рис. 9).

Далее необходимо найти первого студента, имеющего более двух «двоек», т. е. три штуки. Поскольку для расположения студентов, имеющих «двойки», не важен средний балл, нужно отсортировать интересующих нас студентов заново. Для этого с помощью фильтра выберем студентов, имеющих ровно три «двойки», и перенесем их, например, на новый лист. Далее отсортируем строки только по ID.

Видим, что самый маленький ID = 635 (рис 10). Это ответ на второй вопрос.

5. Заключение

На рассмотренных примерах показано, что с помощью электронных таблиц можно успешно решать некоторые задания КЕГЭ, не прибегая к традиционному написанию программ на языках высокого уровня. В этом случае предполагается использование встроенных функций, написание собственных формул, а также применение других средств электронных процессоров, таких как сортировка, фильтрация и т. п. К преимуществам такого метода можно отнести наглядность обрабатываемых данных и скорость получения ответа. Предлагаемый подход является менее затратным во временном контексте подготовки к экзамену, что, однако, ни в коем случае не умаляет ценности и важности владения навыками программирования.

Для отработки навыков решения и самоподготовки рекомендуется использовать материалы, размещенные на сайте К. Ю. Полякова [10], в Открытом банке заданий [8], а также на ресурсе «Демонстрационная версия станции КЕГЭ» [3].

Список источников

- Босова Л. Л. О новых подходах к изучению школьной информатики в условиях цифровой трансформации общества // Информатика в школе. 2022. № 4. С. 5–14. EDN: DKRLZV. DOI: 10.32517/2221-1993-2022-21-4-5-14.
- Босова Л. Л., Куклина И. Д., Мирончик Е. А. Практикум по работе с электронными таблицами в старшей школе // Информатика в школе. 2018. № 10. С. 3–24. EDN: VOSXLR. DOI: 10.32517/2221-1993-2018-17-10-3-24.

3. Демонстрационная версия станции КЕГЭ. <https://kompege.ru>

4. Единый государственный экзамен по информатике // ФИПИ. Демоверсии, спецификации, кодификаторы. <https://fipi.ru/ege/demoversii-specifikacii-kodifikatory#1/tab/151883967-5>

5. Емельянова А. А. Различные способы решения задания № 13 КЕГЭ 2024 на IP адреса и маски // Наука и образование: актуальные вопросы, достижения и инновации. Сборник статей X Международной научно-практической конференции (Пенза, 12 октября 2023 года). Пенза: Наука и Просвещение (ИП Гуляев Г. Ю.), 2023. С. 38–44. EDN: IICUKE.

6. Златопольский Д. М. Выполняем задание 8 ЕГЭ по информатике в среде электронной таблицы // Мир информатики. 2023. № 76. С. 1–8.

7. Златопольский Д. М. Методика выполнения задания 22 демонстрационного варианта ЕГЭ по информатике 2023 года // Мир информатики. 2023. № 75. С. 1–8.

8. Открытый банк тестовых заданий // ФИПИ. <https://ege.fipi.ru/bank>

9. Поляков К. Ю. Динамическое программирование в задачах обработки последовательностей ЕГЭ по информатике // Информатика в школе. 2020. № 5. С. 55–63. EDN: SXZKLG. DOI: 10.32517/2221-1993-2020-19-5-55-63.

10. Поляков К. Ю. Материалы для подготовки к ЕГЭ по информатике (2025). <https://kpolyakov.spb.ru/school/ege.htm>

11. Поляков К. Ю. Программное решение задач ЕГЭ по информатике на IP-адреса и маски подсетей // Информатика в школе. 2024. № 2. С. 73–80. EDN: FQIZOZ. DOI: 10.32517/2221-1993-2024-23-2-73-80.

12. Поляков К. Ю., Еремин Е. А. Информатика. 10 класс (базовый и углубленный уровни). В 2 ч.: учебник. М.: БИНОМ. Лаборатория знаний, 2019. Ч. 1. 352 с.; Ч. 2. 352 с.

13. Поляков К. Ю., Еремин Е. А. Информатика. 11 класс (базовый и углубленный уровни). В 2 ч.: учебник. М.: БИНОМ. Лаборатория знаний, 2019. Ч. 1. 240 с.; Ч. 2. 304 с.

14. Пономарёва Е. А. Электронные таблицы (табличный процессор). Обработка информации в среде табличного процессора: учебно-методическое пособие. Пермь: ИПЦ «Прокрость», 2022. С. 20–25, 29–30.

15. Попов В. С., Алефиренко Е. А., Черницына Л. Ю. Рекурсивная функция для решения задания на динамическое программирование: от беззнания формы компетенции к знанию и пониманию // Информатика в школе. 2024. № 5. С. 85–94. EDN: MDGTEQ. DOI: 10.32517/2221-1993-2024-23-5-85-94.

16. Федеральный государственный образовательный стандарт среднего общего образования. <https://base.garant.ru/70188902/8ef641d3b80ff01d34be16ce9baf6ce0/>

17. Чапурных А. А. Различные способы решения задания № 23 КЕГЭ на перебор вариантов и динамическое программирование // Наука, общество, образование в условиях цифровизации и глобальных изменений. Сборник статей III Междуна-

родной научно-практической конференции (Пенза, 20 августа 2022 года). Пенза: Наука и Просвещение (ИП Гуляев Г. Ю.), 2022. С. 44–48. EDN: LAKNFW.

18. *Шамаева Т. В.* «Динамическое программирование» с помощью MS Excel на примере 23 номера из КЕГЭ по информатике // Архимед: Научно-методический сборник / отв. за

выпуск А. Рубин, Т. Струков, П. Чулков, А. Шевкин. Вып. 18. М.: ООО «МАКС Пресс», 2022. С. 124–136. EDN: RTVFKK. https://www.elemschool.mathedu.ru/text/arhimed_2022_v18/p116/

19. LibreOffice. English documentation. <https://documentation.libreoffice.org/en/english-documentation/>