



# ИТОГОВАЯ АТТЕСТАЦИЯ ПО ИНФОРМАТИКЕ

**В. Г. Смольняков**

*средняя общеобразовательная школа № 17 с углубленным изучением отдельных предметов, г. Щёлково, Московская область, Россия*

## МЕТОДИКА ВЫПОЛНЕНИЯ ЗАДАНИЯ № 8 ЕГЭ ПО ИНФОРМАТИКЕ И ИКТ КОМБИНАТОРНЫМ СПОСОБОМ И СПОСОБОМ ПРОГРАММИРОВАНИЯ НА PYTHON\*

### *Аннотация*

В статье рассмотрена методика выполнения задания № 8 ЕГЭ по информатике и ИКТ двумя способами: математическим комбинаторным расчетом и написанием программы на языке программирования Python. Целью данной методики является успешное выполнение выпускниками заданий, встречающихся под № 8 (до 2021 года — № 10) в Едином государственном экзамене по информатике и ИКТ. Статья имеет междисциплинарный характер, затрагивает вопросы, находящиеся на стыке математики и информатики. Актуальность работы связана с тем, что задания данного типа ежегодно присутствуют в ЕГЭ по информатике и ИКТ, однако успешность выполнения данного задания слишком низкая для заданий базового уровня сложности. Использование средств программирования на ЕГЭ по информатике и ИКТ доступно начиная с 2021 года. Научная новизна работы заключается в применении языка программирования Python для решения заданий такого типа. Особенность методики заключается в поэтапном увеличении сложности алгоритмов и «модульном» применении частей кода, который позволяет использовать «модули» предыдущих задач для решения последующих. Предложены конкретные варианты программ, приведен сравнительный анализ методов для различных прототипов соответствующих заданий. В результате определено, что задание № 8 можно эффективно решать методом программирования.

**Ключевые слова:** информатика, подготовка к ЕГЭ, комбинаторные задачи, программирование.

**DOI:** 10.32517/2221-1993-2021-20-10-54-60

### 1. Введение

Одним из принципов реализации программы среднего общего образования по информатике является реализация межпредметных связей с другими дисциплинами. Контроль качества сформированности таких связей у выпускника осуществляется в соответствии с планируемыми метапредметными результатами. В силу своей специфики информатика наиболее тесно связана с математикой, образуя единую предметную область

«Математика и информатика». В связи с этим логично применение математических методов для решения задач по информатике.

Одним из наиболее ярких представителей «математических» заданий в ЕГЭ по информатике является задание № 8 (до 2021 года это задание было под № 10). Как правило, выпускниками чаще всего используются формулы комбинаторики для выполнения данного задания.

Анализируя качество выполнения задания, можно сказать, что хотя оно и относится к заданиям базового

\* Материалы к статье можно скачать на сайте ИНФО: [http://infojournal.ru/journals/school/school\\_10-2021/](http://infojournal.ru/journals/school/school_10-2021/)

### **Контактная информация**

**Смольняков Вячеслав Геннадьевич**, учитель информатики и математики, средняя общеобразовательная школа № 17 с углубленным изучением отдельных предметов, г. Щёлково, Московская область, Россия; *адрес:* 141107, Россия, Московская область, г. Щёлково, мкрн. Богородский, стр. 11; *e-mail:* vsmolnyakov@yandex.ru

**V. G. Smolnyakov**

School 17, Shchelkovo, Moscow Region, Russia

### **METODOLOGY FOR SOLVING THE TASK NO. 8 OF THE UNIFIED STATE EXAM IN INFORMATICS AND ICT BY A COMBINATORY METHOD AND METHOD OF PROGRAMMING ON PYTHON**

#### **Abstract**

The article discusses the methodology for solving the task No. 8 of the Unified State Exam in informatics and ICT in two ways: by mathematical combinatorial calculation and writing a program in the Python programming language. The purpose of this methodology is the successful completion of task No. 8 (until 2021 — No. 10) in the Unified State Exam in informatics and ICT by graduates. The article is of an interdisciplinary nature, touches upon issues at the intersection of mathematics and informatics. The relevance of the work is due to the fact that tasks of this type are annually present in the Unified State Exam in informatics and ICT, but the success of this task is too low for tasks of the basic level of complexity. The use of programming tools in the Unified State Exam in informatics and ICT is available starting in 2021. The scientific novelty of the work lies in the use of the Python programming language to solve tasks of this type. The peculiarity of the methodology lies in the gradual increase in the complexity of the algorithms and the "modular" application of parts of the code, which allows using the "modules" of previous tasks to solve subsequent ones. Specific versions of the programs are proposed, a comparative analysis of methods for various prototypes of the corresponding tasks is given. As a result, it was determined that task No. 8 can be effectively solved by the programming method.

**Keywords:** informatics, preparation for exam, combinatorial tasks, programming.

уровня сложности, но представляется достаточно трудным для выпускников. На территории Московской области в 2020 году его решили 20 % выпускников, в 2021 году — 48 %, что недостаточно для задания базового уровня сложности.

Основная проблема, которая возникает у выпускников, заключается в том, что из-за большого количества допустимых комбинаций они не могут «в лоб» пересчитать все комбинации, убрать неподходящие. Экзаменуемые используют методы комбинаторики и совершают ошибки в решениях, поскольку в школьной математике данный метод используется достаточно редко и на ЕГЭ по математике не выносятся задания, требующие его применения.

Анализируя указанную проблему, автор пришел к выводу, что процесс можно автоматизировать, отказавшись от расчетов по формулам. С 2021 года ЕГЭ по информатике проводится в компьютерном формате, и выпускник имеет возможность использовать компьютер для выполнения любого задания. Фактически мы имеем дело с простым переборным алгоритмом и счетчиком, учитывающим условия задания. Для данного задания лучше всего подходит язык программирования Python по причинам простоты алгоритма вследствие возможностей языка и его распространенности при изучении информатики.

В статье приведен сравнительный анализ традиционного, комбинаторного способа решения и способа программирования на Python для различных прототипов заданий.

## 2. Теоретическая часть

Решая задание № 10 на ЕГЭ по информатике, выпускники использовали формулы комбинаторики, которые известны им из курса математики. Стоит отметить, что комбинаторика в школьной математике не занимает центрального места и в ЕГЭ по математике комбинаторные формулы для решения заданий не применяются. Поэтому выпускники фактически учат их заново.

Приведем данные формулы:

$$N = K^m. \quad (1)$$

Формула (1) позволяет найти общее количество комбинаций  $N$  длины  $m$ , состоящей из символов алфавита мощностью  $K$ . Данная формула применяется в качестве основной, если все символы алфавита могут повторяться на любых позициях. Если же повторение ограничено, то применяется формула (2):

$$N = K_1 \cdot K_2 \cdot K_3 \cdot \dots \cdot K_m. \quad (2)$$

Формула (2) учитывает количество различных вариантов символов, допустимых на каждой позиции, и позволяет вычислить количество комбинаций  $N$  длины  $m$ , где  $K_1$  — количество вариантов первого символа,  $K_m$  — количество вариантов символа с индексом  $m$ , т. е. последнего в рассматриваемой последовательности.

В задании № 8, как и до этого в задании № 10 по старой нумерации, как правило, требуется нахождение количества элементов выборки, удовлетворяющей требуемым условиям, а не всего количества комбинаций. В этом случае кроме правила произведения, затрагивае-

мого в частных случаях в виде формул (1) и (2), требуется использование правила суммы. Суть правила суммы для рассматриваемого задания состоит в том, что если существует несколько вариантов выбора расположения буквы или цифры и варианты при этом «не пересекаются» (т. е. не существует комбинаций, удовлетворяющих сразу нескольким случаям выбора), то общее количество вариантов будет равно сумме количества вариантов в каждом случае.

В заданиях, где требуется отобрать комбинации, которые удовлетворяют требованиям расположения конкретных символов без точного указания позиций, разумно использовать формулу сочетаний (3):

$$C_n^k = \frac{n!}{(n-k)!k!}. \quad (3)$$

В формуле (3)  $k$  — количество размещаемых символов,  $n$  — количество позиций для размещения. Для решения задачи № 10 ЕГЭ по информатике формул (1)–(3) достаточно.

Однако при трансформации экзамена произошло изменение доступных выпускнику средств выполнения заданий. С 2021 года доступно выполнение данного задания на компьютере. Использование формул комбинаторики обуславливалось неэффективностью перебора комбинаций «в лоб» ввиду их большого количества. Для компьютера количество комбинаций в предлагаемых заданиях не представляет серьезной трудности при вычислениях, и их подсчет не занимает на экзамене значительного времени.

Предлагаемый алгоритм использует особенности языка программирования Python, поэтому построен именно на нем. В частности, используется перебор комбинаций символов через цикл, итерации в котором проводятся непосредственно с элементами, без указания их индексов, т. е. переменная цикла на каждой итерации — это рассматриваемый на данной итерации элемент алфавита.

Суть алгоритма заключается в использовании вложенных циклов, каждый из которых «перебирает» все допустимые варианты конкретного символа. На каждой итерации символы складываются в строку, образуя последовательность, либо удовлетворяющую условию задачи полностью, либо подходящую под часть условий и направляемую на дополнительную проверку условий. Важно, что при использовании алгоритма весь отбор компьютер выполняет на основе задаваемых критериев из условия задачи, т. е. отсутствует необходимость использования комбинаторных формул. Более подробно алгоритм будет раскрыт далее.

## 3. Разбор различных типов заданий

Задания, приведенные в данном разделе, расположены в порядке увеличения сложности алгоритма, т. е. алгоритм каждой следующей задачи использует «модули» предыдущей, поэтому при использовании материалов данной статьи при обучении школьников рекомендуется выдерживать порядок заданий, заданный в статье.

Для каждой задачи дополнительно приведено ее аналитическое решение в виде слайда презентации, полностью эту презентацию можно скачать на сайте ИНФО.

**Задача 1** [1, с. 10].

Игорь составляет таблицу кодовых слов для передачи сообщений, каждому сообщению соответствует свое кодовое слово. В качестве кодовых слов Игорь использует трехбуквенные слова, в которых могут быть только буквы Ш, К, О, Л, А, причем буква К появляется ровно один 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.**

Код программы для решения данной задачи приведен на рисунке 1.

```
count = 0
for c0 in 'ШКОЛА':
    for c1 in 'ШКОЛА':
        for c2 in 'ШКОЛА':
            s = c0 + c1 + c2
            if s.count('K') == 1:
                count += 1
print(count)
```

Рис. 1. Код решения задачи 1 на языке Python

Переменная *count* содержит значение количества подходящих слов. Переменные *c0*, *c1*, *c2* являются переменными циклами, и на каждой итерации их значения являются первой, второй и третьей буквами кодового слова. Далее эти символы складываются в слово, которое проверяется на наличие ровно одной буквы К в слове. Если проверка пройдена успешно, то значение переменной *count*, изначально равное 0, увеличивается на единицу.

Аналитическое решение данной задачи тоже довольно простое, однако в нем необходимо учитывать расположение буквы К и использовать правила суммы и произведения. Аналитическое решение задачи 1 приведено на рисунке 2.



Рис. 2. Аналитическое решение задачи 1

**Задача 2** [4, с. 60].

Сколько существует различных четырехзначных чисел, записанных в восьмеричной системе счисления, в записи которых есть ровно две одинаковые цифры, причем стоящие рядом?

**Решение.**

При решении аналитическим методом необходимо внимательно рассматривать допустимость различных

комбинаций, т. е. сложность аналитического решения возрастает.

Аналитическое решение задачи 2 приведено на рисунке 3.



Рис. 3. Аналитическое решение задачи 2

Сложность программного решения тоже увеличилась, однако новый модуль программы можно использовать в большом количестве задач.

Код программного решения приведен на рисунке 4.

```
count = 0
for c0 in '1234567':
    for c1 in '01234567':
        for c2 in '01234567':
            for c3 in '01234567':
                s = c0 + c1 + c2 + c3
                a = []
                for t in '01234567':
                    a.append(s.count(t))
                a.sort()
                if a[-1] != a[-2] and max(a) == 2:
                    for r in '01234567':
                        if r*2 in s:
                            count += 1
print(count)
```

«сборка чисел»

Подсчёт повторов

Цифры рядом

Рис. 4. Код решения задачи 2 на языке Python с пояснениями

Часть решения до «сборки чисел» уже знакома, стоит отметить ограничение на *c0*, поскольку число не может начинаться с 0, поэтому этого символа нет среди допустимых для переменной *c0*. То есть ограничения на конкретные символы вводятся изначально, и в дальнейшем про это можно не вспоминать. Далее используется список *a*, который служит своего рода «частотным словарем», поскольку в нем отражается количество вхождений того или иного символа. Подобный метод пригодится и в дальнейшем, а в условиях рассматриваемой задачи после сортировки с его помощью проверяется, чтобы самый часто встречающийся символ не встречался столько же раз, сколько другой, и что больше двух раз ни один символ не встречается. Для кодовых слов, удовлетворяющих данному условию, проводится дополнительная проверка, что есть символы, которые встречаются дважды подряд. И только слова, прошедшие такую «двойную» проверку, подходят под условия задачи, увеличивая значение *count* на единицу.

**Задача 3** [3, с. 11].

Саша составляет шестизначные числа, оканчивающиеся на 26, причем цифры в числе не могут повторяться и каждое число содержит или ровно три четные цифры, или ровно две нечетные цифры.

Сколько различных чисел может составить Саша?



**Решение.**

Аналитическое решение этой задачи получается громоздким из-за большого количества налагаемых условий. Более того, чтобы вычислить количество комбинаций, в которых две нечетные цифры, необходимо воспользоваться формулой сочетаний (3), что приводит к необходимости вычислений и риску арифметической ошибки вследствие волнения на экзамене.

Аналитическое решение задачи 3 приведено на рисунке 5, код программного решения — на рисунке 6.

**Задача 3**

Дано:  
шестизначные числа,  
оканчивающиеся на 26  
(\*\*\*\*26)  
Варианты оставшихся цифр:  
0, 1, 3, 4, 5, 7, 8, 9

0 не может стоять первым!  
Цифры не повторяются!

Правило суммы

Три чётные

Вычисления по формулам

Е \* \* \* 2 6 = 2\*5\*4\*3\*1\*1 = 120  
\* Е \* \* 2 6 = 5\*3\*4\*3\*1\*1 = 180  
\* \* Е \* 2 6 = 5\*4\*3\*3\*1\*1 = 180  
\* \* \* Е 2 6 = 5\*4\*3\*3\*1\*1 = 180

5 вариантов  
1, 3, 5, 7, 9

4 варианта  
кроме первой  
цифры

0, 4, 8

660

а

**Задача 3**

Две нечётные

Чтобы определить расположение нечётных цифр, воспользуемся формулой сочетаний

$$C_k^n = \frac{n!}{(n-k)! \cdot k!}$$

k = 2, n = 4, количество сочетаний = 6,  
из них 3 сочетания с первой чётной цифрой

\* N M \* 2 6

4, 8

5 вариантов  
1, 3, 5, 7, 9

4 варианта  
кроме первой  
цифры

0, 4, 8

3 набора

240

Всего: 660 + 360 + 240 = 1260 чисел  
Ответ: 1260

б

**Задача 3**

Дано:  
шестизначные числа,  
оканчивающиеся на 26  
(\*\*\*\*26)  
Варианты оставшихся цифр:  
0, 1, 3, 4, 5, 7, 8, 9

Python 3.9.0 (tags/v3.9.0:4b00757, Oct 5 2020, 11:34:08) [AMD64]  
Type "help", "copyright", "credits" or "license()" for more [>]  
[>>> Evaluate 3.9.0  
[>>> 1260  
[>>> ]

Ответ: 1260

в

Рис. 5. Аналитическое решение задачи 3

При анализе программного решения стоит обратить внимание на допустимые символы. Поскольку цифры не повторяются, а числа заканчиваются на 26, это означает, что на первых четырех позициях среди допустимых символов нет двоек и шестерок (а на первой — нуля). Далее внимания заслуживают модуль проверки повторов (выбираются только те последовательности, где нет повторов цифр, т. е. максимум в массиве *a* равен 1) и по-

```
count = 0
for c0 in '1345789':
    for c1 in '01345789':
        for c2 in '01345789':
            for c3 in '01345789':
                s = c0 + c1 + c2 + c3 + '26'
                a = []
                for t in '0123456789':
                    a.append(s.count(t))
                if max(a) == 1:
                    even = 0
                    odd = 0
                    for t in s:
                        if int(t) % 2 == 0:
                            even += 1
                        else:
                            odd += 1
                    if even == 3 or odd == 2:
                        count += 1
print(count)
```

Проверка, что нет повторов

Подсчёт чётных и нечётных цифр

Рис. 6. Код решения задачи 3 на языке Python с пояснениями

следующий алгоритм расчета четных и нечетных цифр в числе, который и используется в итоге при финальном «отборе» чисел.

**Задача 4** [3, с. 84].

Саша составляет шестибуквенные слова, в которых встречаются только буквы С, О, Н. Причем буква С может стоять только на первом, втором или третьем местах и встречаться или только один раз, или ровно три раза, или не встречаться вовсе. Каждая из других допустимых букв может встречаться в слове на любом месте или не встречаться совсем.

Сколько существует таких слов, которые может написать Саша?

**Решение.**

В целом эта задача не представляет серьезной сложности при любом методе решения. При аналитическом решении важно внимательно рассмотреть все допустимые случаи.

Код программного решения приведен на рисунке 7, аналитическое решение задачи — на рисунке 8.

```
count = 0
for c0 in 'COH':
    for c1 in 'COH':
        for c2 in 'COH':
            for c3 in 'OH':
                for c4 in 'OH':
                    for c5 in 'OH':
                        s = c0 + c1 + c2 + c3 + c4 + c5
                        if s.count('C') == 1 or s.count('C') == 3 \
                           or s.count('C') == 0:
                            count += 1
print(count)
```

Рис. 7. Код решения задачи 4 на языке Python

**Задача 4**

Вычисления по формулам

С буквой «С»

Без буквы «С»

С \* \* \* \* \* = 1\*2\*2\*2\*2\*2=32

\* \* \* \* \* = 2^6 = 64

\* С \* \* \* \* = 2\*1\*2\*2\*2\*2=32

\*\* С \* \* \* = 2\*2\*1\*2\*2\*2=32

CCC \* \* \* = 1\*1\*1\*2\*2\*2=8

О или Н

104

Всего: 104 + 64 = 168 чисел  
Ответ: 168

Рис. 8. Аналитическое решение задачи 4

**Задача 5** [2, № 1928].

Вася составляет 5-буквенные коды из букв М, А, Н, О, К. Каждую букву нужно использовать ровно один раз, при этом код не может начинаться с буквы О и не может содержать сочетания АО.

Сколько различных кодов может составить Вася?

*Решение.*

При аналитическом решении становится очевидно, что проще вычислить количество комбинаций, не удовлетворяющих условию задачи, и вычесть их из общего количества комбинаций, которое можно рассчитать по формуле (2).

Аналитическое решение задачи 5 приведено на рисунке 9, код программного решения — на рисунке 10.

**Задача 5**

В данной задаче проще найти неподходящие случаи и вычесть их количество из общего количества вариантов.

$1 * 1 * 3 * 2 * 1 = 6$

$*** ** = 5 * 4 * 3 * 2 * 1 = 120$

Буквы не повторяются!

$0 * * * * = 1 * 4 * 3 * 2 * 1 = 24$

Всего:  $120 - 24 - 24 = 72$  числа

**Ответ: 72**

$1 * 1 * 3 * 2 * 1 = 6$

АО\*\*\*

\*АО\*\*

\*\*АО\*

\*\*\*АО

$6 * 4 = 24$

Рис. 9. Аналитическое решение задачи 5

```
count = 0
for c0 in 'МАНК':
    for c1 in 'МАНК':
        for c2 in 'МАНК':
            for c3 in 'МАНК':
                for c4 in 'МАНК':
                    s = c0 + c1 + c2 + c3 + c4
                    a = []
                    for t in 'МАНК':
                        a.append(s.count(t))
                    if max(a) == 1 and not 'АО' in s:
                        count += 1
print(count)
```

Выполнение условия, что не начинается с «О»

Проверка, что нет повторов и нет сочетания «АО»

Рис. 10. Код решения задачи 5 на языке Python с пояснениями

Все элементы приведенного решения (рис. 10) уже встречались в предыдущих задачах. Внимания заслуживает стандартный прием для Python, который помогает проверить отсутствие подстроки в собранной строке.

**Задача 6** [5, № 33510].

Тимофей составляет 5-буквенные коды из букв Т, И, М, О, Ф, Е, Й. Буква Т должна входить в код не менее одного раза, а буква Й — не более одного раза.

Сколько различных кодов может составить Тимофей?

*Решение.*

При решении данной задачи необходимо внимательно прочитать условие: не менее одного раза с точки зрения математики интерпретируется как один или более, не более одного — либо не входит вообще, либо входит один раз. Эти варианты и нужно рассмотреть при аналитическом решении.

Код программного решения приведен на рисунке 11, аналитическое решение задачи — на рисунке 12.

Все рассмотренные выше задачи принадлежат к одному типу заданий № 8, посвященному подсчету коли-

```
count = 0
for c0 in 'ТИМОФЕЙ':
    for c1 in 'ТИМОФЕЙ':
        for c2 in 'ТИМОФЕЙ':
            for c3 in 'ТИМОФЕЙ':
                for c4 in 'ТИМОФЕЙ':
                    s = c0 + c1 + c2 + c3 + c4
                    if s.count('Й') <= 1 and s.count('Т') >= 1:
                        count += 1
print(count)
```

Рис. 11. Код решения задачи 6 на языке Python

**Задача 6**

Вычисление по формулам

Не более одного раза – либо не встречаются, либо только один раз  
Не менее одного раза – встречаются один и более раз

$*** ** = 6 * 6 * 6 * 6 * 6 - 5 * 5 * 5 * 5 * 5 = 4651$

Й не входит

$Й * * * * = [(1 * 6 * 6 * 6 * 6) - (1 * 5 * 5 * 5 * 5)] * 5 = 3355$

Й входит 1 раз  
5 наборов

Й \* \* \* \*    \* \* Й \* \*  
\* Й \* \* \*    \* \* \* Й \*

Всего:  $4651 + 3355 = 8006$  чисел

**Ответ: 8006**

Рис. 12. Аналитическое решение задачи 6

чества различных комбинаций цифр или букв. Однако существует второй тип заданий, в котором необходимо определить номер кодового слова по порядку, либо кодовое слово по его номеру. Решить такую задачу тоже можно и аналитическим, и программным способами.

**Задача 7** [4, с. 204].

Все 4-буквенные слова, составленные из букв И, Л, О, С, Т, записаны в алфавитном порядке и пронумерованы, начиная с 1. Начало списка:

1. ИИИИ
2. ИИИЛ
3. ИИИО
4. ИИИС
5. ИИИТ
6. ИИЛИ

Под каким номером стоит слово «СИТО»?

*Решение.*

Аналитическое решение, как правило, основано на переходе к системе счисления с основанием, равным мощности алфавита. Стоит не забывать, что номеру 1 соответствует кодовое слово 0000, т. е. имеется «сдвиг» номера от его «значения» на 1. Программное решение позволяет циклом «дойти» до нужного слова и просто посмотреть его номер, используя отдельную переменную-счетчик.

Код решения приведен на рисунке 13.

Фактически в этом решении используется тот же принцип перебора, что и в предыдущих задачах. Единственное,

```
n = 0
for c0 in 'ИЛОСТ':
    for c1 in 'ИЛОСТ':
        for c2 in 'ИЛОСТ':
            for c3 in 'ИЛОСТ':
                s = c0 + c1 + c2 + c3
                n += 1
                if s == 'СИТО':
                    print(n)
```

Рис. 13. Код решения задачи 7 на языке Python

на что принципиально нужно обратить внимание: порядок букв в строках подстановки для переменных  $c_0$  —  $c_3$  должен быть таким же, как последняя буква в первых кодовых словах (как правило, совпадает с алфавитным).

Аналитическое решение задачи 7 приведено на рисунке 14.

**Задача 7** Вычисление по формулам

Алгоритм:  
И = 0, Л = 1, О = 2, С = 3, Т = 4  
(пятеричная система счисления)

Номер слова на 1 больше, чем его «числовое» значение

$СИТО = 3042_5 = 3 \cdot 5^3 + 0 \cdot 5^2 + 4 \cdot 5^1 + 2 \cdot 5^0 = 375 + 20 + 2 = 397 + 1 = 398$

**Ответ: 398**

Рис. 14. Аналитическое решение задачи 7

### Задача 8 [2, № 2790].

Петя составляет список из 4-буквенных слов, в состав которых входят только буквы О, С, Е, Н, Ъ. Петя расположил слова в обратном алфавитном порядке. Начало списка:

1. ЪЪЪЪ
2. ЪЪЪС
3. ЪЪЪО
4. ЪЪЪН
5. ЪЪЪЕ
6. ЪЪСЪ

Какое слово стоит в этом списке под номером 100?

*Решение.*

Это задача, обратная к задаче 7, и она аналитически решается похожим образом. Алгоритм программного решения аналогичен предыдущей задаче, но вместо слова в условии проверяется номер.

Код программного решения приведен на рисунке 15, аналитическое решение задачи — на рисунке 16.

```
number = 0
for c0 in 'ЬСОНЕ':
    for c1 in 'ЬСОНЕ':
        for c2 in 'ЬСОНЕ':
            for c3 in 'ЬСОНЕ':
                s = c0 + c1 + c2 + c3
                number += 1
                if number == 100:
                    print(s)
```

Рис. 15. Код решения задачи 8 на языке Python

**Задача 8** Вычисление по формулам

Алгоритм:  
Ь = 0, С = 1, О = 2, Н = 3, Е = 4  
(пятеричная система счисления)

Номер слова на 1 больше, чем его «числовое» значение

$100 - 1 = 99_{10} = 344_5 = 0344_5 = ЪНЕЕ$

**Ответ: ЪНЕЕ**

Рис. 16. Аналитическое решение задачи 8

Существуют задачи, относящиеся к данному типу, однако с большим количеством условий. Такая задача разобрана ниже.

### Задача 9 [5, № 35466].

Вероника составляет 3-буквенные коды из букв В, Е, Р, О, Н, И, К, А, причем буква В должна входить в код ровно один раз. Все полученные коды Вероника записала в алфавитном порядке и пронумеровала. Начало списка выглядит так:

1. ААВ
2. АВА
3. АВЕ

На каком месте будет записан первый код, не содержащий ни одной буквы А?

*Решение.*

Аналитическое решение данной задачи через системы счисления невозможно, поскольку некоторые буквенные комбинации не являются допустимыми, поэтому не входят в последовательность кодовых слов. Поэтому нумерация «сбивается» и сдвиг постепенно увеличивается. Таким образом, вследствие короткой длины кодового слова наиболее простым аналитическим решением будет перебор вариантов. Первой подходящей комбинацией является комбинация «ВЕЕ», для которой и необходимо записать порядковый номер.

Аналитическое решение задачи 9 приведено на рисунке 17.

**Задача 9** Вычисление по формулам

С учётом длины кода и поставленного условия проще всего вычислить «в лоб», т.е. перебирая все возможные варианты

**А, В, Е, И, К, Н, О, Р**

1. ААВ	7. АВО	13.АОВ	19.ВАН
2. АВА	8. АВР	14.АРВ	20.ВАО
3. АВЕ	9. АЕВ	15.ВАА	21.ВАР
4. АВИ	10.АИВ	16.ВАЕ	22.ВЕА
5. АВК	11.АКВ	17.ВАИ	23.ВЕЕ
6. АВН	12.АНВ	18.ВАК	

**Ответ: 23**

Рис. 17. Аналитическое решение задачи 9

Программное решение основано на прежних принципах. При переборе стоит не забывать, что в алфавите присутствует восемь букв, которые в строке подстановки следует располагать в алфавитном порядке. После создания текущей комбинации она проверяется на условия задачи. Важно, что в программном решении реализован именно отбор по условиям задачи, а не поиск комбинации «ВЕЕ», что упрощает аналитическую подготовку программного решения.

Код программы приведен на рисунке 18.

```
number = 0
a = []
for c0 in 'АВЕИKNOP':
    for c1 in 'АВЕИKNOP':
        for c2 in 'АВЕИKNOP':
            s = c0 + c1 + c2
            if s.count('B') == 1:
                number += 1
                if not 'A' in s:
                    a.append(number)
print(min(a))
```

Рис. 18. Код решения задачи 9 на языке Python



#### 4. Сравнительный анализ

Основными критериями эффективности решения на экзамене для выпускника являются время, потраченное на решение, и уровень сложности (что прямо пропорционально вероятности допустить ошибку).

Для простых задач, таких как задачи 1, 4, 5 (и отчасти 7, 8), аналитический метод не уступает программному. Для более сложных задач метод программного решения эффективнее, поскольку позволяет не формулировать ограничения самостоятельно на основе условия, а брать их оттуда напрямую в код ветвления.

#### 5. Заключение

В статье рассмотрена методика выполнения задания № 8 ЕГЭ по информатике и ИКТ методом программирования, которая сравнивается с традиционными аналитическими методами решения задания.

Сравнивая методы решения рассмотренных задач, можно сделать вывод, что задание № 8 можно эффективно

решать методом программирования. Основных конструкций несколько, и они понятны и легко запоминаемы. Такой метод позволяет легко учитывать ограничения, накладываемые в условиях задачи. Однако стоит учитывать, что, если выпускники изучают другие языки программирования, а не Python, возможности этих языков могут ограничить экзаменующихся в программном решении.

#### Список использованных источников

1. Демонстрационная версия ЕГЭ по информатике 2021 года.  
<https://fipi.ru/ege/demoversii-specifikacii-kodifikatory#!tab/151883967-5>
2. ЕГЭ по информатике // Сайт К. Ю. Полякова. <https://kpolyakov.spb.ru/school/ege.htm>
3. Информатика и ИКТ. Подготовка к ЕГЭ-2021. 10 тренировочных вариантов по демоверсии 2021 года: учебное пособие / под ред. С. Ю. Кулабухова. Ростов н/Д.: Легион, 2021. 160 с.
4. Крылов С. С., Чуркина Т. Е. ЕГЭ. Информатика и ИКТ: типовые экзаменационные варианты: 20 вариантов. М.: Национальное образование, 2021. 256 с.
5. Решу ЕГЭ. Информатика. <https://inf-ege.sdangia.ru/>

## НОВОСТИ

### Правительство Российской Федерации утвердило стратегическое направление в области цифровой трансформации образования

Правительство Российской Федерации утвердило стратегическое направление в области цифровой трансформации образования, относящейся к сфере деятельности Министерства просвещения Российской Федерации. Соответствующее распоряжение подписал председатель Правительства России Михаил Мишустин. Документ опубликован на официальном интернет-портале правовой информации.

Стратегическое направление утверждается до 2030 года. Оно охватывает все уровни общего образования, а также среднего профессионального образования.

В ходе реализации стратегического направления в систему образования будут внедрены такие технологии, как искусственный интеллект в части рекомендательных систем и интеллектуальных систем поддержки принятия решений, большие данные в части использования методов интеллектуального анализа значительных объемов информации для поддержки принятия управленческих решений и повышения качества, системы распределенного реестра, облачные технологии. Также предусмотрено создание условий для функционирования электронной

информационно-образовательной среды. Появятся сервисы «Цифровой помощник ученика», «Цифровой помощник родителя», «Цифровой помощник учителя», «Цифровое портфолио ученика», «Библиотека цифрового образовательного контента».

Сервисы позволят использовать современный верифицированный цифровой образовательный контент, реализовывать программы углубленного уровня, выстраивать индивидуальные траектории обучения, а также повышать уровень профессиональной компетентности педагогов. Также они дадут возможность автоматизировать проверку домашних заданий с помощью искусственного интеллекта, снизят административную нагрузку на учителей, обеспечат возможность формировать цифровое портфолио обучающегося для подачи в колледжи или вуз, принимать управленческие решения на основе больших данных.

Отмечается, что целью цифровой трансформации является обеспечение эффективной информационной поддержки участников образовательных отношений в рамках процесса получения образования и управления образовательной деятельностью.

(По материалам ТАСС и федерального портала «Российское образование»)