



В. С. Попов

Тверской государственный университет, г. Тверь, Россия;

Московский городской педагогический университет, г. Москва, Россия;

Московский государственный технический университет имени Н. Э. Баумана (национальный исследовательский университет), г. Москва, Россия

ФОРМИРОВАНИЕ КОМПЕТЕНЦИЙ СТАРШЕКЛАССНИКОВ В ОБЛАСТИ ПРОГРАММИРОВАНИЯ: РЕШЕНИЕ ЗАДАНИЯ 25 ЕГЭ ПО ИНФОРМАТИКЕ С МАСКАМИ ЧИСЕЛ*

Аннотация

В статье приведены возможные решения задания 25 ЕГЭ по информатике, в условии которого содержатся маски чисел. Рассмотренные методы решения заданий включают, во-первых, широко используемый вызов функции `fnmatch` модуля `fnmatch` для проверки соответствия числа заданной маске, во-вторых, использование операций над строками для проверки соответствия маске, в-третьих, перебор диапазонов чисел в циклах для чисел одной разрядности с последующей проверкой соответствия маске отдельных разрядов чисел. Предложены возможные перспективные задания формата задания 25 повышенного уровня сложности, требующие проверки дополнительных условий, включая условия для четности некоторых цифр числа, для суммы цифр числа, для количества различных цифр числа, для количества нетривиальных делителей числа. Описаны дополнительные возможности функции `fnmatch`, приведен соответствующий пример задания и программного кода. Для представленных заданий и различных видов их решений приведены формируемые компетенции старшеклассников в области программирования. Показано, что при формировании компетенций в области программирования на описанном предметном содержании также формируются математические компетенции. Формируемые компетенции обучающихся входят в ключевые компетенции модели *Key competences for lifelong learning* как цифровые компетенции и STEM-компетенции.

Ключевые слова: компетенция, компетентность, ЕГЭ по информатике, перебор в цикле, делимость, маска числа, `fnmatch`.

Информация об авторе

Попов Владислав Сергеевич, аспирант Института педагогического образования и социальных технологий, Тверской государственной университет, г. Тверь, Россия; эксперт, Институт развития профильного обучения, Московский городской педагогический университет, г. Москва, Россия; старший преподаватель кафедры «Информационные системы и телекоммуникации», факультет «Информатика и системы управления», Московский государственный технический университет имени Н. Э. Баумана (национальный исследовательский университет), г. Москва, Россия; *e-mail:* popov_vlad@mail.ru

* Материалы к статье можно скачать на сайте ИНФО: http://infojournal.ru/journals/school/school_02-2025/

V. S. Popov

Tver State University, Tver, Russia;

Moscow City University, Moscow, Russia;

Bauman Moscow State Technical University, Moscow, Russia

DEVELOPING OF HIGH SCHOOL STUDENTS' PROGRAMMING COMPETENCES: SOLVING TASK 25 OF THE UNIFIED STATE EXAM IN INFORMATICS WITH NUMBER MASKS

Abstract

The article presents possible solutions to tasks 25 of the Unified State Exam in Informatics containing number masks. The considered methods for solving tasks include, firstly, the widely used call of the `fnmatch` function of the `fnmatch` module to check whether a number matches a given mask, secondly, using of string operations to check whether a number matches a mask, and thirdly, enumerating number ranges in cycles for numbers of the same digit capacity with subsequent checking of individual digits of numbers for compliance with the mask. Possible promising task 25 of the Unified State Exam in Informatics format of an advanced complexity level are proposed, requiring checking additional conditions, including conditions for: the parity of some digits of a number, the sum of digits of a number, the number of different digits of a number, the number of non-trivial divisors of a number. Additional capabilities of the `fnmatch` function are described, a corresponding example of a task and program code are given. For the presented tasks and various types of their solutions, the developed competences of high school students in the field of programming are presented. It is shown that when developing competences in the field of programming on the described subject content, mathematical competences are also developed. The developed competences of students are included in the Key competences for lifelong learning framework as digital competences and STEM competences.

Keywords: competence, competency, USE in informatics, looping through a range, divisibility, number mask, `fnmatch`.

Information about the author

Vladislav S. Popov, postgraduate student, Institute of Pedagogical Education and Social Technology, Tver State University, Tver, Russia; Expert, Institute for Development of Profile Education, Moscow City University, Moscow, Russia; Senior Lecturer, Department of Information Systems and Telecommunications, Faculty "Computer Science and Control Systems", Bauman Moscow State Technical University, Moscow, Russia; e-mail: popov_vlad@mail.ru

1. Введение

Задание 25 ЕГЭ по информатике было обновлено в 2021 году в связи с внедрением компьютерной формы ЕГЭ по информатике (КЕГЭ) [7, 8, 14]. В соответствии со спецификациями контрольных измерительных материалов для проведения единого государственного экзамена по информатике за 2021–2025 годы проверяемым элементом содержания является «умение создавать собственные программы (10–20 строк) для обработки целочисленной информации» [18]. Несмотря на неизменный проверяемый элемент содержания, задачи, предлагаемые в задании 25, можно условно разделить на два типа, в данной статье рассматривается второй тип:

- 1) задачи, в которых требуется определить делители числа [19, с. 347]; содержатся в демонстрационных вариантах КИМ ЕГЭ по информатике 2021, 2022, 2025 годов;
- 2) задачи, содержащие маску числа; включены в демонстрационные варианты КИМ ЕГЭ по информатике 2023–2025 годов.

С 2022 года успешное выполнение задания 25 приносит экзаменуемому один первичный балл [1]. Процент решения задания 25 по результатам ЕГЭ по информатике за 2021–2024 годы приведен в таблице.

Таблица

**Процент выполнения задания 25
ЕГЭ по информатике за 2021–2024 годы**

Год	2021	2022	2023	2024
Процент выполнения	28,6	24,8	42,5	22,6

Представленные данные свидетельствуют о том, что задание 25 выполняют лишь около четверти выпускников. Только в 2023 году с данным заданием справились 42,5 % выпускников — во многом за счет использования представленного ниже решения на языке программирования Python, содержащего вызов функции *fnmatch*. Хотя по данным, доступным на сайте К. Ю. Полякова [9], задание данного типа содержалось в материалах ЕГЭ по информатике также в 2022 году, к 2023 году экзаменуемые смогли выучить методы решения этого нового задания. Снижение количества обучающихся, справившихся с заданием 25 в 2024 году, вызвано большим количеством задач на вычисление делителей чисел [9]. Можно отметить, что в 2022 году выпускники столкнулись с риском появления задания нового типа [11, с. 329], в то время как в 2024 году — с рисками усложнения задания и возврата к заданию 25 старого типа [11, с. 328–329].

Ю. В. Григорьев, К. Н. Фадеева, С. Я. Санжикова из Чувашского государственного педагогического

университета в собственном исследовании показали, что задания КИМ ЕГЭ по информатике, решаемые методами программирования, в том числе задание 25, входят в число самых сложных для выполнения обучающимися [3, с. 181]. К аналогичному выводу в отношении задания 25 приходят Е. Н. Гусева и С. А. Повитухин [4]. Ю. В. Розова, Е. А. Редько в качестве первого результата собственного исследования также указывают на необходимость формирования и развития навыков практического программирования [14, с. 263]. М. А. Сорочинский и М. М. Белолубский отмечают, что около половины заданий экзамена можно решить как с помощью написания программ в системах программирования, так и посредством табличных редакторов, из языков программирования для ЕГЭ по информатике выделяют Python [17, с. 115–116].

Актуальность представленного в данной статье материала обоснована описанной выше сложностью задания 25 ЕГЭ по информатике и других связанных с программированием заданий для обучающихся, возможным усложнением задания [10, с. 4], отсутствием в научно-педагогической литературе описания методов решения данного задания и формируемых компетенций обучающихся, необходимостью развития таких компетенций обучающихся, которые позволят им реализовать собственный потенциал и перейти на следующий уровень образования посредством получения высоких результатов на ЕГЭ по информатике.

2. Способы решения задания 25 с маской числа из демонстрационных версий ЕГЭ по информатике

2.1. Задание 25 ЕГЭ по информатике демоверсии 2023 года

Рассмотрим способы решения задания демонстрационной версии ЕГЭ по информатике 2023 года [5].

Задание 1.

Назовем маской числа последовательность цифр, в которой также могут встречаться следующие символы:

- символ «?» означает ровно одну произвольную цифру;
- символ «*» означает любую последовательность цифр произвольной длины; в том числе «*» может задавать и пустую последовательность.

Например, маске 123*4?5 соответствуют числа 123405 и 12300405.

Среди натуральных чисел, не превышающих 10^{10} , найдите все числа, соответствующие маске 1?2139*4, делящиеся на 2023 без остатка. В ответе запишите в первом столбце таблицы все найденные числа в порядке возрастания, а во втором столбце — соответствующие им

```
import fnmatch
for n in range(0, 10 ** 10, 2023):
    if fnmatch.fnmatch(str(n), '1?2139*4'):
        print(n, n // 2023)
```

Листинг 1. Решение задания 25 демонстрационной версии ЕГЭ по информатике 2023 года в Python с использованием функции fnmatch

```
for n in range(0, 10 ** 10, 2023):
    s = str(n)
    if s[0] == '1' and s[2:6] == '2139' and s[-1] == '4':
        print(n, n // 2023)
```

Листинг 2. Решение задания 25 демонстрационной версии ЕГЭ по информатике 2023 года в Python с использованием индексации и последующего сравнения строк

результаты деления этих чисел на 2023. Количество строк в таблице для ответа избыточно.

В листинге 1 приведен программный код для решения задания с использованием функции *fnmatch* [16, с. 276]. В первой строке подключен модуль *fnmatch*, одноименная функция вызвана как *fnmatch.fnmatch*.

Проверка первого условия — соответствия числа маске *1?2139*4* — выполняется с использованием функции *fnmatch* модуля *fnmatch*. Первым аргументом функции *fnmatch* является число, преобразованное в строковый тип данных с использованием функции *str*, вторым аргументом функции *fnmatch* является маска. Функция возвращает значение *True*, если число *n* соответствует маске, и *False* в противном случае.

Выполнение второго условия, содержащегося в задании, — делимость числа на 2023 — гарантируется перебором чисел от 0 до 10^{10} с шагом 2023. Первое число в перебираемой последовательности чисел является нулем, который делится на 2023 без остатка. Шаг 2023 и ноль как первое число перебираемой последовательности гарантируют, что все последующие числа также кратны 2023. Число 10^{10} не включается в перебираемый диапазон, так как оно не соответствует заданной маске.

В результате выполнения алгоритм выведет верный ответ — следующие четыре строки:

```
162139404 80148
1321399324 653188
1421396214 702618
1521393104 752048
```

В первом столбце содержится подходящее под условия задания число *n*, во втором столбце — частное от деления *n* на 2023.

В листинге 2 показан способ проверки соответствия числа *n* маске без использования функции *fnmatch*.

В листинге 2 перебор чисел осуществляется аналогично перебору чисел в листинге 1, но проверка соответствия маске выполнена с использованием строкового типа данных и индексации строк [2, с. 122–123]: первый символ строки имеет нулевой индекс и проверяется как: *s[0] == '1'*, символы подстроки «2139» имеют индексы 2, 3, 4, 5, соответствующее условие записано с использованием среза списка: *s[2:6] == '2139'*, последний символ числа проверяется с использованием отрицательного индекса: *s[-1] == '4'*. Проверку подстроки с использованием среза списка *s[2:6]* можно заменить на

```
for n in range(1021394, 1921394 + 1, 100000):
    if n % 2023 == 0:
        print(n, n // 2023)
for n in range(10213904, 19213994 + 1, 10):
    if n % 2023 == 0:
        if (n // 100) % 10000 == 2139:
            print(n, n // 2023)
for n in range(102139004, 192139994 + 1, 10):
    if n % 2023 == 0:
        if (n // 1000) % 10000 == 2139:
            print(n, n // 2023)
for n in range(1021390004, 1921399994 + 1, 10):
    if n % 2023 == 0:
        if (n // 10000) % 10000 == 2139:
            print(n, n // 2023)
```

Листинг 3. Решение задания 25 демонстрационной версии ЕГЭ по информатике 2023 года в Python с использованием отдельных циклов для перебора чисел различной разрядности, операций целочисленного деления и получения остатка от целочисленного деления для проверки подстроки «2139» в нужной позиции

несколько проверок отдельных символов, в этом случае условие будет записано в виде:

```
s[0] == '1' and s[2] == '2' and s[3] == '1' and
s[4] == '3' and s[5] == '9' and s[-1] == '4'.
```

Знание, умение и опыт практического применения индексации строк, включая применение среза списка и отрицательных индексов, могут заменить использование функции *fnmatch*.

В листинге 3 приведен программный код, в котором осуществляется отдельный перебор для чисел различной разрядности: существуют отдельные циклы для перебора семизначных, восьмизначных, девятизначных и десятизначных чисел. Числа с большим количеством разрядов превышают 10^{10} .

В первом цикле представленного в листинге 3 программного кода выполняется перебор всего 10 чисел — от 1021394 до 1921394 включительно с шагом 100000. В данном диапазоне чисел символ «?» маски заменяется на цифру, символ «*» маски соответствует пустой последовательности. Поскольку все перебираемые числа соответствуют маске, в теле цикла производится только проверка делимости числа n на 2023. Если число делится на 2023 без остатка, оно выводится на экран вместе с частным от деления на 2023. Второй, третий и четвертый циклы программного кода перебирают числа в диапазонах [10213904, 19213994], [102139004, 192139994], [1021390004, 1921399994] соответственно с шагом 10. Поскольку в каждом цикле проверяются числа одинаковой разрядности (одинаковой длины), начинающиеся на 1 и заканчивающиеся на 4, в теле цикла следует проверить делимость чисел на 2023 и равенство определенных разрядов чисел значению 2139. В листинге 3 такая проверка осуществлена при помощи операций целочисленного деления и получения остатка от целочисленного деления. Например, при $n = 10213904$ значение выражения $(n // 100)$ равняется 102139, а результатом следующей операции получения остатка от целочисленного деления на 10000 будет число 2139.

2.2. Задание 25 ЕГЭ по информатике демоверсии 2024 года

Задание 25 демонстрационной версии ЕГЭ по информатике 2024 года почти полностью копирует аналогичное задание демоверсии 2023 года:

```
from fnmatch import fnmatch
for n in range(0, 10 ** 10, 2024):
    if fnmatch(str(n), '1?2157*4'):
        print(n, n // 2024)
```

Листинг 4. Решение задания 25 демонстрационной версии ЕГЭ по информатике 2024 года в Python с использованием функции *fnmatch*

```
for n in range(0, 10 ** 10, 2024):
    s = str(n)
    if s[0] == '1' and s[2] == '2' and s[3] == '1' and s[4] == '5' and s[5] == '7'
    and s[-1] == '4':
        print(n, n // 2024)
```

Листинг 5. Решение задания 25 демонстрационной версии ЕГЭ по информатике 2024 года в Python с использованием индексации и последующего сравнения строк

Задание 2.

Среди натуральных чисел, не превышающих 10^{10} , найдите все числа, соответствующие маске $1?2157*4$, делящиеся на 2024 без остатка. В ответе запишите в первом столбце таблицы все найденные числа в порядке возрастания, а во втором столбце — соответствующие им результаты деления этих чисел на 2024 [6].

Решения задания с использованием функции *fnmatch* и операций со строками, соответствующие листингам 1 и 2, приведены в листингах 4 и 5. В первой строке листинга 4 из модуля *fnmatch* импортирована функция *fnmatch*, в этом случае она вызывается только по имени: *fnmatch(str(n), '1?2157*4')*.

В результате выполнения программного кода был получен верный ответ на задание:

```
142157664 70236
1021575544 504731
1121571264 554136
1221577104 603546
1321572824 652951
1421578664 702361
1521574384 751766
1621570104 801171
1721575944 850581
1821571664 899986
1921577504 949396
```

3. Дополнительные возможности функции *fnmatch*

Шаблон функции *fnmatch*, задаваемый вторым аргументом, кроме описанных символов «*» и «?» также может содержать другие специальные последовательности [15, с. 303; 21]:

- символ «*» соответствует любой, в том числе пустой, последовательности символов;
- символ «?» соответствует ровно одному любому символу;
- «[seq]» соответствует одному из символов, включенных в последовательность seq;
- «![seq]» соответствует одному символу, не входящему в последовательность seq.

Пример усложненного задания для изучения дополнительных возможностей функции *fnmatch* приведен в задании 3.

Задание 3.

Среди натуральных чисел, не превышающих 10^{10} , найдите все числа, соответствующие маске `1?2157*4`, делящиеся на 2024 без остатка, для которых четность второй и третьей цифр с конца числа различна. В ответе запишите в первом столбце таблицы все найденные числа в порядке возрастания, а во втором столбце — соответствующие им результаты деления этих чисел на 2024.

Вариант решения данного задания, использующий дополнительные возможности функции *fnmatch*, приведен в листинге 6.

В программе (листинг 6) для каждого числа проверяются три условия:

- `fnmatch(str(n), '1?2157[02468]4')` — условие того, что на месте символа «*» указана ровно одна четная цифра, занимающая предпоследний разряд числа и следующая после нечетной цифры 7;
- `fnmatch(str(n), '1?2157*[02468][!02468]4')` — условие того, что третий с конца числа разряд является четной цифрой, второй с конца числа разряд является нечетной цифрой, перед данными двумя разрядами слева могут присутствовать дополнительные цифры числа;
- `fnmatch(str(n), '1?2157*[13579][!13579]4')` — условие того, что третий с конца числа разряд является нечетной цифрой, второй с конца числа разряд является четной цифрой, перед данными двумя разрядами слева могут присутствовать дополнительные цифры числа.

Если одно из представленных условий является истинным, программа выводит число и частное от его деления на 2024.

```
from fnmatch import fnmatch
for n in range(0, 10 ** 10, 2024):
    if (fnmatch(str(n), '1?2157[02468]4') or
        fnmatch(str(n), '1?2157*[02468][!02468]4') or
        fnmatch(str(n), '1?2157*[13579][!13579]4')):
        print(n, n // 2024)
```

Листинг 6. Решение усложненного задания 25 демонстрационной версии ЕГЭ по информатике 2024 года в Python.
Добавленное условие: четность второй и третьей цифр с конца числа различна

```
from fnmatch import fnmatch
for n in range(0, 10 ** 10, 2024):
    if fnmatch(str(n), '1?2157*4'):
        n_copy = n
        total = 0
        while n_copy > 0:
            total += n_copy % 10
            n_copy //= 10
        if total > 40:
            print(n, n // 2024)
```

Листинг 7. Решение усложненного задания 25 демонстрационной версии ЕГЭ по информатике 2024 года в Python.
Добавленное условие: сумма цифр числа превышает 40

Ответом на данное задание будут шесть строк:

```
1021575544 504731
1221577104 603546
1521574384 751766
1621570104 801171
1721575944 850581
1921577504 949396
```

Еще одним возможным решением является сравнение четности указанных символов, в таком случае условие оператора *if* может быть записано в виде:

```
fnmatch(str(n), '1?2157*4') and int(str(n)
[-3]) % 2 != int(str(n)[-2]) % 2
```

4. Примеры задания 25 ЕГЭ по информатике повышенного уровня сложности и соответствующие решения

Модифицированное задание с маской может включать дополнительные условия, например, ограничение на сумму цифр числа:

Задание 4.

Среди натуральных чисел, не превышающих 10^{10} , найдите все числа с суммой цифр больше 40, соответствующие маске `1?2157*4`, делящиеся на 2024 без остатка. В ответе запишите в первом столбце таблицы все найденные числа в порядке возрастания, а во втором столбце — соответствующие им результаты деления этих чисел на 2024.

Вариант решения данного задания с использованием функции *fnmatch* приведен в листинге 7. Для решения задания использован классический алгоритм вычисления суммы цифр в числе: пока число *n_copy* больше нуля, значение переменной *total* увеличивается на очередную

```
total = 0
for char in str(n):
    total += int(char)
```

Листинг 8. Вычисление суммы цифр числа n с использованием итерирования по символам строки

цифру числа ($n_copy \% 10$), после чего выполняется целочисленное деление переменной n_copy на 10 для перехода к следующей цифре числа.

Возможно вычисление суммы цифр числа с использованием итерирования по символам строки (листинг 8).

По сравнению с результатом, полученным для задания 25 демонстрационной версии ЕГЭ по информатике 2024 года, вывод программы содержит только четыре строки:

```
1421578664 702361
1721575944 850581
1821571664 899986
1921577504 949396
```

Другим примером дополнительного условия в задании является условие присутствия в числе указанного количества различных цифр.

Задание 5.

Среди натуральных чисел, не превышающих 10^{10} , найдите все числа, соответствующие маске $1?2157*4$, делящиеся на 2024 без остатка, *имеющие более 6 различных цифр в своей записи*. В ответе запишите в первом столбце таблицы все найденные числа в порядке возрастания, а во втором столбце — соответствующие им результаты деления этих чисел на 2024.

Решение данного задания с использованием множества приведено в листинге 9. В случае дублирования цифр в числе множество s хранит внутри себя каждую цифру только один раз, поэтому вычисление мощности множества $len(s)$ позволяет определить количество различных цифр в числе.

Вывод данной программы содержит шесть строк:

```
1321572824 652951
1421578664 702361
1521574384 751766
1621570104 801171
1821571664 899986
1921577504 949396
```

Еще одним примером дополнительного условия в задании является условие присутствия в числе указанного количества четных или нечетных цифр.

Задание 6.

Среди натуральных чисел, не превышающих 10^{10} , найдите все числа, соответствующие маске $1?2157*4$, делящиеся на 2024 без остатка, *имеющие более половины нечетных цифр в своей записи*. В ответе запишите в первом столбце таблицы все найденные числа в порядке возрастания, а во втором столбце — соответствующие им результаты деления этих чисел на 2024.

Решение представлено в листинге 10.

```
from fnmatch import fnmatch
for n in range(0, 10 ** 10, 2024):
    if fnmatch(str(n), '1?2157*4'):
        n_copy = n
        s = set()
        while n_copy > 0:
            s.add(n_copy % 10)
            n_copy //= 10
        if len(s) > 6:
            print(n, n // 2024)
```

Листинг 9. Решение усложненного задания 25 демонстрационной версии ЕГЭ по информатике 2024 года в Python. Добавленное условие: число должно содержать более 6 различных цифр

```
from fnmatch import fnmatch
for n in range(0, 10 ** 10, 2024):
    if fnmatch(str(n), '1?2157*4'):
        odd_cnt = 0
        for c in str(n):
            if c in '13579':
                odd_cnt += 1
        if odd_cnt > len(str(n)) / 2:
            print(n, n // 2024)
```

Листинг 10. Решение усложненного задания 25 демонстрационной версии ЕГЭ по информатике 2024 года в Python. Добавленное условие: число должно иметь более половины нечетных цифр в своей записи

В отличие от программного кода для решения двух предыдущих заданий, в данном коде (листинг 10) вычисление количества нечетных цифр выполняется в цикле с посимвольным перебором числа, представленного в виде строки.

Ответом на приведенное задание являются шесть пар чисел:

```
1021575544 504731
1121571264 554136
1221577104 603546
1521574384 751766
1721575944 850581
1921577504 949396
```

Следует отметить, что для четырех предыдущих модифицированных вариантов задания 25 демонстрационного варианта ЕГЭ по информатике 2024 года довольно легко применить добавленное условие самостоятельно, без использования дополнительного программного кода, поскольку ответ на исходное задание 25 содержит 11 пар чисел, из которых впоследствии следует отобрать подходящие строки. Однако в случаях, когда таких строк десятки и сотни или аналитическое вычисление является слишком сложным (например, как это будет показано в следующем задании), дополнительный программный код становится необходимым для успешного решения задания.

Возможно объединение двух типов задания 25 ЕГЭ по информатике и присутствие в одной задаче как условий для маски числа, так и условий для его делителей.

Задание 7.

Среди натуральных чисел, не превышающих 10^{10} , найдите все числа, соответствующие маске $1?2157^*4$, делящиеся на 2024 без остатка, имеющие не менее 100 нетривиальных делителей. В ответе запишите в первом столбце таблицы все найденные числа в порядке возрастания, во втором столбце — соответствующие им результаты деления этих чисел на 2024, в третьем столбце — количество нетривиальных делителей числа. *Примечание:* тривиальными делителями числа являются единица и само число.

Программный код для решения данного задания представлен в листинге 11. Функция `number_of_factors(n)` вычисляет количество нетривиальных делителей числа n . В цикле тела функции `number_of_factors(n)` выполняется перебор возможных делителей от 2 до квадратного корня из n включительно. Для каждого найденного делителя числа n , кроме квадратного корня числа n , имеется парный делитель числа n , больший квадратного корня из n . Примерами парных делителей для числа 100 являются: 2 и 50, 4 и 25, 5 и 20. Именно поэтому при нахождении делителя счетчик количества делителей `cnt` увеличивается на 2. После цикла происходит проверка того, что квадратный корень из n является целым числом: `n ** 0.5 == int(n ** 0.5)`. Если данное условие истинно, то существует натуральный делитель числа n , равный квадратному корню числа n , и значение счетчика `cnt` уменьшается на единицу. Для проверки условия существования не менее 100 нетривиальных делителей числа описанная функция вызывается в конце программного кода, представленного в листинге 11: `cnt = number_of_factors(n)`, после чего происходит проверка условия: `cnt >= 100`. Описанная функция будет корректно и быстро работать даже для больших аргументов n .

Вывод программы, приведенной в листинге 11, будет ответом на представленное задание:

```
142157664 70236 142
1121571264 554136 166
1621570104 801171 446
1721575944 850581 158
1821571664 899986 158
1921577504 949396 190
```

5. Формируемые компетенции обучающихся в области программирования

Под компетенциями понимают «интегрированный набор знаний, навыков и деятельностных установок, которые мобилизуются в определенном контексте для решения определенной задачи, для достижения опре-

```
def number_of_factors(n):
    cnt = 0
    for divider in range(2, int(n ** 0.5) + 1):
        if n % divider == 0:
            cnt += 2
    if n ** 0.5 == int(n ** 0.5):
        cnt -= 1
    return cnt
for n in range(0, 10 ** 10, 2024):
    s = str(n)
    if s[0] == '1' and s[2] == '2' and s[3] == '1' and s[4] == '5' and s[5] == '7'
       and s[-1] == '4':
        cnt = number_of_factors(n)
        if cnt >= 100:
            print(n, n // 2024, cnt)
```

Листинг 11. Решение усложненного задания 25 демонстрационной версии ЕГЭ по информатике 2024 года в Python. Добавленное условие: количество делителей числа не менее 100

деленного результата» [20, с. 38]. При этом «ключевым вопросом развития универсальных компетентностей является вопрос соотношения с предметными (дисциплинарными) знаниями и навыками» [20, с. 20]. Для реализации компетентного подхода в школе, актуализировавшегося с появлением федеральных рабочих программ [13], следует выделить основные компоненты формируемых компетенций старшеклассников в рамках рассмотренного предметного содержания. В качестве формируемых компетенций в области программирования при обучении решению рассмотренных заданий с масками чисел можно выделить:

- перебор чисел в цикле с заданным шагом в указанном диапазоне;
- перебор цифр числа;
- обработка отдельных разрядов чисел с использованием как мультипликативных операций, так и строк, включая операцию индексации строк, индексацию строк слева и справа, использование среза списка;
- работа с различными типами данных;
- использование структур данных на примере строк и множеств;
- определение функций;
- использование модулей на примере модуля *fnmatch*.

Изучение модуля *fnmatch* и одноименной функции может служить пропедевтикой к знакомству с более сложным содержанием — регулярными выражениями, широко используемыми при решении задания 24 ЕГЭ по информатике на обработку строковой информации [2, с. 131–137].

Кроме компетенций в области программирования на рассмотренном предметном содержании обучающиеся формируют компетенции в области математики, используя операции целочисленного деления и получения остатка от целочисленного деления для получения различных разрядов чисел, изучая делимость чисел и перебирая числа в арифметической прогрессии посредством цикла. По классификации модели ключевых компетенций Key competences for lifelong learning компетенции в области программирования и математики можно отнести к ключевым компетенциям — цифровым и STEM-компетенциям [12].

6. Заключение

В статье рассмотрены различные способы решения вариантов задания 25 ЕГЭ по информатике, содержащих маски числа, приведены примеры усложненных заданий и соответствующие программные коды на языке Python для их решения. Показаны решения с использованием функции *fnmatch*, индексации строк, операций целочисленного деления и получения остатка от целочисленного деления. Приведены формируемые компетенции обучающихся, включающие не только компетенции в области программирования, но и математические компетенции.

Список источников

1. Акимова И. В., Байбекова С. Н. Новые тенденции в ЕГЭ по информатике в 2021–22 учебном году // Преподава-

ние информационных технологий в Российской Федерации. Сборник научных трудов материалов Двадцатой открытой Всероссийской конференции (Москва, 19–20 мая 2022 года). М.: 1С-Паблишинг, 2022. С. 357–358. EDN: XTQJZT.

2. Гильдин А. Г., Зайдулина С. Г., Ахтямов Н. Т. Методика выполнения заданий ЕГЭ по информатике на основе знаний об обработке строк на языке Python // Педагогический журнал Башкортостана. 2023. № 2 (100). С. 120–138. EDN: GXDJVL. DOI: 10.21510/18173292_2023_100_2_120_138.

3. Григорьев Ю. В., Фадеева К. Н., Санжикова С. Я. Анализ результатов пробного Единого государственного экзамена по информатике // Вестник науки. 2023. Т. 3. № 10 (67). С. 177–182. EDN: YBKKNM.

4. Гусева Е. Н., Повитухин С. А. Использование языков программирования Python и PascalABC.NET в задачах ЕГЭ по информатике // Исследование различных направлений современной науки. Сборник материалов VII международной очно-заочной научно-практической конференции (Москва, 14 сентября 2022 года). Т. 1. М.: Научно-издательский центр «Империя», 2022. С. 43–52. EDN: HGWNSV.

5. Демонстрационный вариант контрольных измерительных материалов единого государственного экзамена 2023 года по информатике. М.: Федеральный институт педагогических измерений, 2022. 22 с.

6. Демонстрационный вариант контрольных измерительных материалов единого государственного экзамена 2024 года по информатике. М.: Федеральный институт педагогических измерений, 2023. 24 с.

7. Князькова О. В. Анализ изменений, внесенных в ЕГЭ 2021 по информатике // Информатика и прикладная математика. 2020. № 26. С. 65–71. EDN: BNQGOV.

8. Лантеева З. А. Подготовка школьников к компьютерному ЕГЭ по информатике: из опыта работы // Цифровизация современной школы. Сборник материалов IV областного семинара (Киров, 28–30 июня 2021 года). Киров: Кировское региональное отделение Межрегиональной Ассоциации учителей и преподавателей информатики, 2021. С. 46–52. EDN: APXYRP.

9. Поляков К. Ю. ЕГЭ по информатике: подготовка к ЕГЭ–2025 по информатике, разбор задач ЕГЭ–2025 по информатике, материалы для подготовки к ЕГЭ. <https://kpolyakov.spb.ru/school/ege.htm>

10. Поляков К. Ю., Еремин Е. А. Как нам реорганизовать ЕГЭ по информатике? // Информатика в школе. 2019. № 3 (146). С. 2–7. EDN: WXNBVV. DOI: 10.32517/2221-1993-2019-18-3-2-7.

11. Попов В. С., Абросимова-Романова Л. А. Анализ рисков выпускников и выделение необходимых компетенций в связи с изменением среднего процента выполнения заданий ЕГЭ по информатике за 2019–2023 годы и введением компьютерного формата экзамена // Новые информационные технологии в образовании. Сборник научных трудов XXIV Международной научно-практической конференции (Москва, 30–31 января 2024 года). М.: 1С-Паблишинг, 2024. С. 326–330. EDN: RJIRBK.

12. Попов В. С., Абросимова-Романова Л. А. Трансформация ключевых компетенций модели Key competences for lifelong learning // Текущие вызовы в подготовке кадров. Обучение специалистов по современным направлениям информационных технологий, кибербезопасности и ИКТ-электроники, актуальным для экономики данных. Сборник научных трудов конференции «Преподавание информационных технологий в Российской Федерации» (Тверь, 16–17 мая 2024 года). Тверь: Тверской государственный университет, 2024. С. 474–477. EDN: QUFFBF.

13. Попов В. С., Алефиренко Е. А., Черницына Л. Ю. Компетентностный подход в федеральных рабочих программах по информатике на уровне среднего общего образования // Текущие вызовы в подготовке кадров. Обучение специалистов по современным направлениям информационных технологий, кибербезопасности и ИКТ-электроники, актуальным для экономики данных. Сборник научных трудов конференции «Преподавание информационных технологий в Российской Федерации» (Тверь, 16–17 мая 2024 года). Тверь: Тверской государственный университет, 2024. С. 604–607. EDN: BUWEMV.

14. Розова Ю. В., Редько Е. А. Особенности подготовки к компьютерному ЕГЭ по информатике и ИКТ // Современная наука и образование: новые подходы и актуальные исследования. Материалы II Всероссийской научно-практической конференции (Чебоксары, 11 мая 2021 года). Чебоксары: Негосударственное образовательное частное учреждение дополнительного профессионального образования «Экспертно-методический центр», 2021. С. 253–264. EDN: VIPGBV.

15. Россум Г., Дрейк Ф. Л. Дж., Откидач Д. С., Задка М., Левис М., Монтаро С., Реймонд Э. С., Кучлинг А. М., Лембург М.-А., Йи К.-П., Ксиллаг Д., Петрилли Х. Г., Варсав Б. А., Ахлстром Дж. К., Роскинд Дж., Шеменор Н., Мулендер С. Язык программирования Python: пер. с англ. Д. С. Откидач. 2001. 454 с.

16. Соколова А. Н., Шалагинова Н. В. Использование практико-ориентированных задач при обучении старшеклассников программированию на Python // Проблемы теории и практики инновационного развития и интеграции современной науки и образования. Материалы IV Международной междисциплинарной конференции (Москва, 15 февраля 2023 года). М.: Государственный университет просвещения, 2024. С. 276–281. EDN: BIINXG.

17. Сорочинский М. А., Белолобский М. М. Подготовка к ЕГЭ по информатике и ИКТ: обзор заданий и решение задач

на основе языка программирования Python // Международный научно-исследовательский журнал. 2021. № 8-3 (110). С. 114–117. EDN: OJTQPX. DOI: 10.23670/IRJ.2021.110.8.097.

18. Спецификация контрольных измерительных материалов для проведения в 2025 году единого государственного экзамена по информатике. М.: Федеральный институт педагогических измерений, 2024. 12 с.

19. Трекина А. С., Богданова М. В. Методические аспекты подготовки учащихся к ЕГЭ по информатике с использованием языка программирования Python // Актуальные проблемы методики обучения информатике и математике в современной школе. Материалы Международной научно-практической интернет-конференции (Москва, 18–24 апреля 2022 года) / под ред. Л. Л. Босовой, Д. И. Павлова. М.: МПГУ, 2022. С. 345–352. EDN: WEOGQT.

20. Универсальные компетентности и новая грамотность: от лозунгов к реальности: научное издание: монография / под ред. М. С. Добряковой, И. Д. Фрумина; при участии К. А. Баранникова, Н. Зиила, Дж. Мосс, И. М. Реморенко, Я. Хаутамяки; Нац. исслед. ун-т «Высшая школа экономики». М.: Изд. дом Высшей школы экономики, 2020. 472 с. EDN: ADLXUC. DOI: 10.17323/978-5-7598-2177-9.

21. fnmatch — Unix filename pattern matching // Python 3 Documentation. <https://docs.python.org/3/library/fnmatch.html>