



DOI: 10.32517/2221-1993-2023-22-2-63-72

В. К. Маркелов, О. А. Завьялова

Шуйский филиал Ивановского государственного университета, г. Шуя, Ивановская область, Россия

ЯЗЫК ПРОГРАММИРОВАНИЯ PYTHON КАК АЛЬТЕРНАТИВНЫЙ ИНСТРУМЕНТ ДЛЯ РЕШЕНИЯ ЗАДАНИЙ ЕГЭ ПО ИНФОРМАТИКЕ

Аннотация

Программирование составляет основу содержательной линии «Алгоритмы и программирование», которая изучается на протяжении всего школьного курса информатики, поэтому задачи по программированию являются одним из тематических блоков единого государственного экзамена по предмету «Информатика». В частности, одним из учебных языков, используемых в рамках решения задач ЕГЭ по информатике, является язык программирования Python. Данный язык может использоваться как для решения заданий раздела «Программирование», так и для решения других заданий единого государственного экзамена, упрощая способ их решения при наличии у обучающихся навыков составления программ на данном языке программирования с использованием основных алгоритмических структур (следование, ветвление, повторение) и структур данных (массивы, строки).

В статье рассматриваются возможности использования языка программирования Python в качестве альтернативного инструмента для решения заданий № 2, № 5, № 15 и № 16 ЕГЭ по информатике на примере заданий из демонстрационного варианта контрольных измерительных материалов ЕГЭ 2023 года, а также заданий из открытого банка заданий ЕГЭ.

Ключевые слова: информатика, программирование, Python, единый государственный экзамен.

1. Введение

В 2022 году информатика вошла в тройку предметов — лидеров для сдачи единого государственного экзамена. Согласно аналитическим данным Рособнадзора, количество желающих сдать информатику в 2022 году выросло более чем на 30 000, составив 128 044 человек, в 2021 году таких выпускников было 95 198, а в 2020 году — 83 610 [1]. В соответствии с письмом Рособнадзора от 28.01.2021 № 10-21 «О направлении

разъяснений по отдельным вопросам, связанным с организацией работы конфликтных и предметных комиссий по учебному предмету “Информатика и информационно-коммуникационные технологии (ИКТ)”» [7], начиная с 2021 года ЕГЭ по информатике проводится в компьютерной форме (КЕГЭ). При этом с каждым годом возрастает сложность экзаменационных заданий. Поэтому подготовка к ЕГЭ по информатике является актуальной задачей как для обучающихся, так и для учителей информатики.

Контактная информация

Маркелов Валерий Константинович, магистрант 2-го года обучения по направлению подготовки 44.04.01 «Педагогическое образование» (профиль «Информационные технологии в профессиональной деятельности педагога»), Шуйский филиал Ивановского государственного университета, г. Шуя, Ивановская область, Россия; *адрес:* 155908, Россия, Ивановская область, г. Шуя, ул. Кооперативная, д. 24; *e-mail:* v.a.l.e.m.a.r.k@yandex.ru

Завьялова Ольга Алексеевна, канд. пед. наук, доцент кафедры математики, информатики и методики обучения, Шуйский филиал Ивановского государственного университета, г. Шуя, Ивановская область, Россия; *адрес:* 155908, Россия, Ивановская область, г. Шуя, ул. Кооперативная, д. 24; *e-mail:* ooolga30@yandex.ru

V. K. Markelov, O. A. Zavyalova

Shuya branch of Ivanovo State University, Shuya, Ivanovo Region, Russia

THE PYTHON PROGRAMMING LANGUAGE AS AN ALTERNATIVE TOOL FOR SOLVING TASKS OF THE UNIFIED STATE EXAM IN INFORMATICS

Abstract

Programming is the basis of the content line "Algorithms and programming", which is studied throughout the entire school course of informatics, therefore programming tasks are one of the thematic blocks of the Unified State Exam (USE) in informatics. In particular, one of the educational languages used in solving tasks of the USE in informatics is the Python programming language. This language can be used both for solving the problems of the "Programming" section, and for solving other tasks of the Unified State Exam, simplifying the way to solve them if the students have the skills to write programs in this programming language using the main algorithmic structures (sequencing, selection, and iteration) and data structures (arrays, strings).

The article describes the possibilities of using the Python programming language as an alternative tool for solving tasks No. 2, No. 5, No. 15 and No. 16 of the USE in informatics using the example of tasks from the demo version of the USE 2023 control measuring materials, as well as tasks from the open bank of USE tasks.

Keywords: informatics, programming, Python, Unified State Exam.

Программирование составляет основу содержательной линии «Алгоритмы и программирование», изучаемой на протяжении всего школьного курса информатики, поэтому задачи по программированию являются одним из тематических блоков ЕГЭ по информатике [4, с. 69]. Для решения таких заданий могут использоваться учебные языки программирования, изучаемые в школе, одним из которых является Python.

Python — это мощный профессиональный язык программирования, который благодаря своему простому синтаксису позволяет обучающимся быстро освоить базовые алгоритмические структуры (следование, ветвление, повторение) и акцентировать основное внимание на оттачивании навыков программирования [3, с. 96]. Он преднамеренно оптимизирован для ускорения разработки — его встроенный инструментальный набор предоставляет программистам возможность разрабатывать программы за гораздо меньшее время, чем то, которое пришлось бы потратить, применяя ряд других инструментов [2, с. 43].

В соответствии с Федеральным государственным образовательным стандартом основного общего образования [9] язык программирования Python является одним из учебных языков программирования, рекомендуемых как на базовом, так и на углубленном уровнях изучения информатики. К причинам выбора данного языка для обучения школьников также можно отнести особенность структуры программы на языке Python, которая заставляет обучающихся соблюдать правила отступов, что делает программу более понятной и легко читаемой, а также наличие в языке высокоуровневых структур, таких как строки, списки, множества, словари, которые вместе с обширным набором библиотечных функций для их обработки позволяют решать сложные задачи уже на раннем уровне обучения программированию [6, с. 226]. Изучение Python в старших классах также обусловлено интересом к нему на более высокой ступени обучения — в российских и зарубежных университетах, популярностью в индустрии информационных технологий, о чем свидетельствует и достаточное количество онлайн-курсов, предлагающих научиться программировать на Python и новичка, и специалиста [5, с. 48].

Согласно методическим рекомендациям по подготовке и проведению единого государственного экзамена по учебному предмету «Информатика и информационно-коммуникационные технологии (ИКТ)» в компьютерной форме в 2022 году к письму Рособнадзора от 31.01.2022 № 04-18 «О направлении методических документов, рекомендуемых при организации и проведении государственной итоговой аттестации по образовательным программам основного общего образования и среднего общего образования в 2022 году», для выполнения экзаменационных заданий каждому участнику предоставляется автоматизированное рабочее место без выхода в сеть Интернет с установленным специализированным ПО «Станция КЕГЭ», набором стандартного ПО (текстовые редакторы, редакторы электронных таблиц, среды программирования на языках: Школьный алгоритмический язык, C#, C++, Pascal, Java, Python) [8, с. 1095].

Поэтому учебный язык программирования Python может использоваться как для решения задач по про-

граммированию, так и для решения других заданий ЕГЭ по информатике, упрощая способ их решения при наличии у обучающихся навыков составления программ на данном языке программирования с использованием основных алгоритмических структур (следование, ветвление, повторение) и структур данных (массивы, строки).

В частности, к таким экзаменационным заданиям относятся:

- задание № 2 (умение строить таблицы истинности и логические схемы);
- задание № 5 (формальное исполнение простого алгоритма, записанного на естественном языке, или умение создавать линейный алгоритм для формального исполнителя с ограниченным набором команд, или умение восстанавливать исходные данные линейного алгоритма по результатам его работы);
- задание № 15 (знание основных понятий и законов математической логики);
- задание № 16 (вычисление рекуррентных выражений).

2. Возможности решения заданий ЕГЭ по информатике с использованием языка программирования Python

Рассмотрим указанные типы заданий более подробно на примере заданий из демонстрационного варианта КИМ ЕГЭ 2023 года по информатике, а также на примере заданий из открытого банка заданий ЕГЭ (<http://ege.fipi.ru/bank>). Демонстрационный вариант КИМ ЕГЭ по информатике доступен для скачивания на сайте Федерального института педагогических измерений (ФИПИ) [10] по следующей ссылке: https://doc.fipi.ru/ege/demoversii-specifikacii-kodifikatory/2023/inf_11_2023.zip

2.1. Задание № 2. Построение таблиц истинности и логических схем

Одним из заданий ЕГЭ, решение которого можно упростить при помощи программы на языке программирования Python, является задание № 2, направленное на проверку умения строить таблицы истинности и логические схемы. В данном задании необходимо определить, какому столбцу таблицы истинности логической функции F соответствует каждая из переменных w , x , y и z при условии, что дан лишь частично заполненный фрагмент таблицы. Соответствующее задание представлено на рисунке 1.

Чтобы упростить решение данного задания, можно написать программу на языке Python для построения полностью заполненного фрагмента таблицы истинности логической функции F :

$$F = \neg(w \rightarrow x) \vee (y \rightarrow z) \vee \neg y.$$

Наиболее важным этапом при написании вспомогательной программы к заданию № 2 является вычисление значения логической функции, которое может осуществляться при помощи логических операций: инверсии (логическое отрицание), дизъюнкции (логическое сложение) и конъюнкции (логическое умножение), причем ре-

Миша заполнял таблицу истинности логической функции F

$$\neg(w \rightarrow x) \vee (y \rightarrow z) \vee \neg y,$$

но успел заполнить лишь фрагмент из трёх различных её строк, даже не указав, какому столбцу таблицы соответствует каждая из переменных w, x, y, z .

				F
	0			0
	1	0		0
		1	0	0

Определите, какому столбцу таблицы соответствует каждая из переменных w, x, y, z .

В ответе напишите буквы w, x, y, z в том порядке, в котором идут соответствующие им столбцы (сначала буква, соответствующая первому столбцу; затем буква, соответствующая второму столбцу, и т.д.). Буквы в ответе пишите подряд, никаких разделителей между буквами ставить не нужно.

Пример. Функция F задана выражением $\neg x \vee y$, зависящим от двух переменных, а фрагмент таблицы имеет следующий вид.

		F
0	1	0

В этом случае первому столбцу соответствует переменная y , а второму столбцу – переменная x . В ответе следует написать: yx .

Рис. 1. Задание № 2 из открытого банка заданий ЕГЭ по информатике

зультат вычисления дизъюнкции и конъюнкции в Python может быть получен при помощи встроенных функций вычисления максимума и минимума соответственно.

В свою очередь, более сложные логические операции — импликация (логическое следование) и эквиваленция (логическая равнозначность) — могут выражаться как через инверсию, дизъюнкции и конъюнкции, так и через операции отношения: «меньше или равно», «равно». Возможности реализации логических операций на базе языка программирования Python для вычисления значения логической функции представлены в таблице на примере двух логических переменных x и y .

Программа к заданию № 2 из открытого банка заданий ЕГЭ по информатике представлена на рисунке 2. Данная программа перебирает все возможные наборы значений переменных w, x, y, z , используя алгоритми-

ческую конструкцию «цикл с параметром», вычисляет значение логической функции F и выводит на экран те наборы значений переменных, при которых логическая функция является ложной (т. е. принимает значение False или 0).

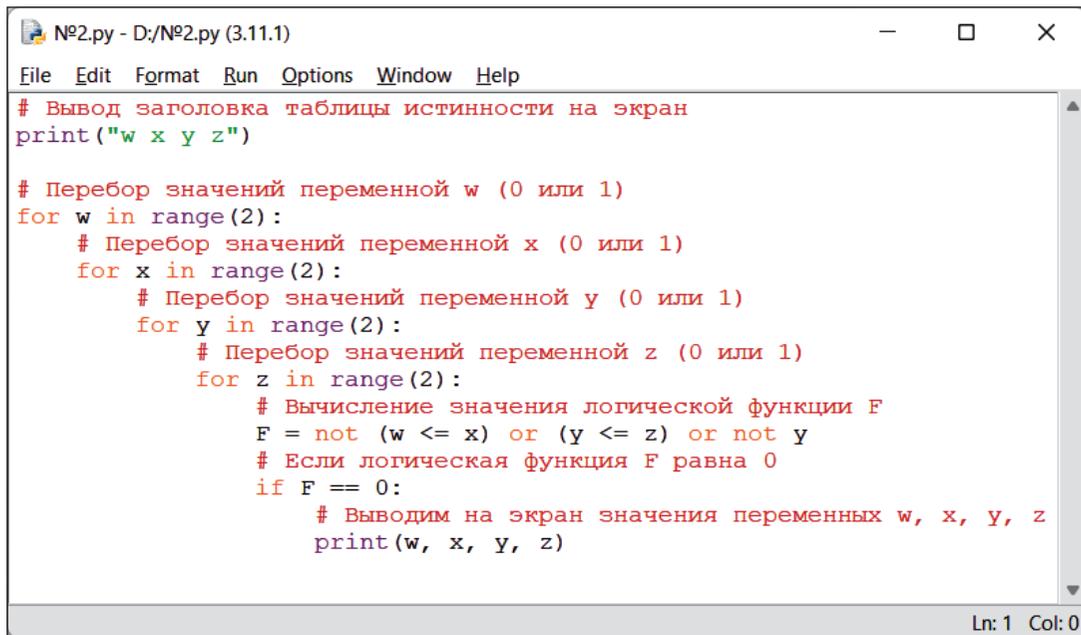
Результат работы программы представлен на рисунке 3. Данный фрагмент таблицы полностью совпадает с фрагментом таблицы, который представлен в задании, за исключением порядка строк и столбцов таблицы. Следовательно, для решения задания обучающемуся достаточно сопоставить полученную таблицу с таблицей, которая представлена в задании, и получить ответ.

В данном случае в полученном фрагменте таблицы строки записаны в обратном порядке. Первый столбец — это столбец y , так как известно, что в остальных столбцах таблицы переменная принимает либо значе-

Таблица

Реализация логических операций на базе языка программирования Python

Логическая операция	Логическое выражение в Python	
	Запись с использованием логических операторов	Запись с использованием встроенных функций Python и операций отношения
Инверсия ($\neg x$)	(not x)	
Дизъюнкция ($x \vee y$)	(x or y)	max (x, y)
Конъюнкция ($x \wedge y$)	(x and y)	min (x, y)
Импликация ($x \rightarrow y$)	(not x or y)	(x <= y)
Эквиваленция ($x \leftrightarrow y$)	((not x and y) or (x and not y))	(x == y)



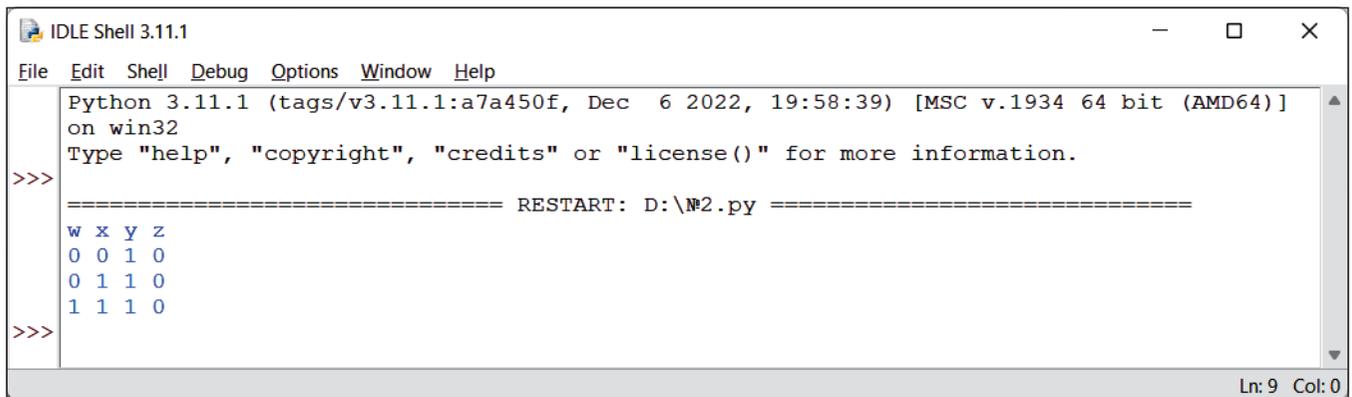
```

№2.py - D:\№2.py (3.11.1)
File Edit Format Run Options Window Help
# Вывод заголовка таблицы истинности на экран
print("w x y z")

# Перебор значений переменной w (0 или 1)
for w in range(2):
    # Перебор значений переменной x (0 или 1)
    for x in range(2):
        # Перебор значений переменной y (0 или 1)
        for y in range(2):
            # Перебор значений переменной z (0 или 1)
            for z in range(2):
                # Вычисление значения логической функции F
                F = not (w <= x) or (y <= z) or not y
                # Если логическая функция F равна 0
                if F == 0:
                    # Выводим на экран значения переменных w, x, y, z
                    print(w, x, y, z)
Ln: 1 Col: 0

```

Рис. 2. Программа к заданию № 2 ЕГЭ по информатике



```

IDLE Shell 3.11.1
File Edit Shell Debug Options Window Help
Python 3.11.1 (tags/v3.11.1:a7a450f, Dec 6 2022, 19:58:39) [MSC v.1934 64 bit (AMD64)]
on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:\№2.py =====
w x y z
0 0 1 0
0 1 1 0
1 1 1 0
>>>
Ln: 9 Col: 0

```

Рис. 3. Результат работы программы к заданию № 2 ЕГЭ по информатике

ние 0, либо значения 0 и 1. В свою очередь, чтобы логическая функция F являлась ложной (т. е. принимала значение False или 0), переменная y должна принимать значение 1 во всех строках фрагмента таблицы. Второй столбец — это столбец x , так как в первой и во второй строках таблицы переменная x принимает значения 0 и 1 соответственно. Тогда третий столбец — это столбец w , так как во второй строке таблицы при $x = 1$ переменная w принимает значение 0. Следовательно, оставшийся четвертый столбец — это столбец z .

Таким образом, ответ к заданию: $uxwz$.

2.2. Задание № 5. Работа с линейными алгоритмами для формального исполнителя

Следующее экзаменационное задание может быть полностью решено при помощи программы на языке Python. Таким заданием является задание № 5, ориентированное на проверку умения создавать линейный алгоритм для формального исполнителя с ограниченным набором команд, а также умения восстанавливать исходные данные линейного алгоритма по результатам его работы. В соответствующем задании демонстрационного

варианта КИМ дается алгоритм, который в результате обработки некоторого натурального числа N получает натуральное число R . При этом обучающемуся необходимо найти минимальное число N , после обработки которого с помощью заданного алгоритма получается число R , большее 40. Данное задание представлено на рисунке 4.

Для решения этого задания можно написать программу на языке Python, которая перебирает все натуральные числа N от 1 до 40 (поскольку в результате обработки алгоритма должно получиться число, большее 40) и для каждого из чисел применяет алгоритм, представленный в задаче:

- 1) Строится двоичная запись числа N .
- 2) Далее эта запись обрабатывается по следующему правилу:
 - а) если сумма цифр в двоичной записи числа четная, то к этой записи справа дописывается 0, а затем два левых разряда заменяются на 10;
 - б) если сумма цифр в двоичной записи числа нечетная, то к этой записи справа дописывается 1, а затем два левых разряда заменяются на 11.

На вход алгоритма подаётся натуральное число N . Алгоритм строит по нему новое число R следующим образом.

1. Строится двоичная запись числа N .
2. Далее эта запись обрабатывается по следующему правилу:
 - а) если сумма цифр в двоичной записи числа чётная, то к этой записи справа дописывается 0, а затем два левых разряда заменяются на 10;
 - б) если сумма цифр в двоичной записи числа нечётная, то к этой записи справа дописывается 1, а затем два левых разряда заменяются на 11.

Полученная таким образом запись является двоичной записью искомого числа R .

Например, для исходного числа $6_{10} = 110_2$ результатом является число $1000_2 = 8_{10}$, а для исходного числа $4_{10} = 100_2$ результатом является число $1101_2 = 13_{10}$.

Укажите **минимальное** число N , после обработки которого с помощью этого алгоритма получается число R , большее 40. В ответе запишите это число в десятичной системе счисления.

Рис. 4. Задание № 5 из демонстрационного варианта КИМ ЕГЭ по информатике

3) Полученная таким образом запись является двоичной записью искомого числа R .

Программа к заданию № 5 демонстрационного варианта КИМ ЕГЭ по информатике представлена на рисунке 5.

Данная программа перебирает исходные натуральные числа N при помощи цикла с параметром и выполняет для каждого из них заданный в задаче алгоритм:

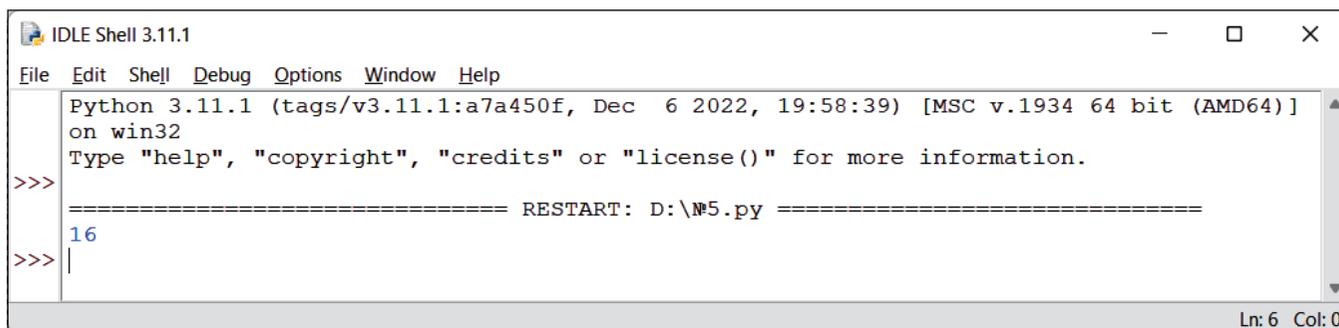
1) Строится двоичная запись N_2 числа N при помощи встроенной функции $bin(N)$, при этом получается стро-

```

№5.py - D:\№5.py (3.11.1)
File Edit Format Run Options Window Help
# Перебор исходных чисел N от 1 до 40 (не включая 41)
for N in range(1, 41):
    # Переводим число N2 в двоичную систему счисления в формате
    # 0b10101, где 10101 - число в двоичной системе счисления
    N2 = bin(N)
    # Удаляем первые два символа "0b" из числа N2 -
    # получаем число в двоичной системе счисления
    N2 = N2[2:]
    # Подсчитываем сумму цифр в двоичной записи числа
    # (количество единиц в записи числа)
    S = N2.count("1")
    # Если сумма цифр в двоичной записи числа четная
    if S % 2 == 0:
        # Два левых разряда числа заменяются на 10 и
        # к записи числа справа дописывается 0
        N2 = "10" + N2[2:] + "0"
    # Если условие не выполняется
    else:
        # Два левых разряда числа заменяются на 11 и
        # к записи числа справа дописывается 1
        N2 = "11" + N2[2:] + "1"
    # Полученное число N2 - двоичная запись числа R
    # Переведем это число в десятичную систему, чтобы получить число R
    R = int(N2, 2)
    # Если в результате работы алгоритма получается число больше 40
    if R > 40:
        # Выводим на экран исходное число N
        print(N)
        # Поскольку первое выведенное число N будет минимальным числом,
        # выходим из цикла, чтобы не появлялись другие числа
        break
Ln: 1 Col: 0

```

Рис. 5. Программа к заданию № 5 ЕГЭ по информатике



```

IDLE Shell 3.11.1
File Edit Shell Debug Options Window Help
Python 3.11.1 (tags/v3.11.1:a7a450f, Dec 6 2022, 19:58:39) [MSC v.1934 64 bit (AMD64)]
on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:\№5.py =====
16
>>>
Ln: 6 Col: 0

```

Рис. 6. Результат работы программы к заданию №5 ЕГЭ по информатике

ка вида «0bXXXXXX», где XXXXXX — число N в двоичной системе счисления. Из полученной двоичной записи $N2$ удаляются первые два символа «0b» при помощи команды:

```
N2 = N2[2:]
```

2) Для дальнейшей обработки двоичной записи $N2$ подсчитывается сумма цифр S в ней. Для этого в Python используется метод *count*, который позволяет определить количество символов в строке.

3) При помощи условия:

```
S % 2 == 0
```

осуществляется проверка, является ли сумма цифр S четной (сумма цифр S является четной, если она делится на 2 без остатка).

а) Если условие выполняется, то к двоичной записи $N2$ справа дописывается 0, а затем два левых разряда заменяются на 10. То есть двоичная запись $N2$ будет иметь вид:

```
"10" + N2[2:] + "0"
```

(запись начинается с символов «10», затем следует двоичная запись $N2$ без первых двух символов, а в конце записи дописывается «0»).

б) Если условие не выполняется, то к двоичной записи $N2$ справа дописывается 1, а затем два левых разряда заменяются на 11. То есть двоичная запись $N2$ будет иметь вид:

```
"11" + N2[2:] + "1"
```

(запись начинается с символов «11», затем следует двоичная запись $N2$ без первых двух символов, а в конце записи дописывается «0»).

4) Полученная двоичная запись $N2$ — это двоичная запись числа R , следовательно, чтобы получить число R , необходимо перевести число $N2$ в десятичную систему счисления при помощи команды:

```
R = int(N2, 2)
```

5) При помощи условия $R > 40$ осуществляется проверка, является ли число, полученное в результате работы алгоритма, больше 40. Если данное условие выполняется, на экран выводится искомое число N . Данное число будет искомым минимальным числом, поэтому необходимо завершить перебор чисел (выйти из цикла) при помощи команды *break*. Если этого не сделать, на экран выведутся все числа N от 1 до 40, для которых в результате работы алгоритма получается число R , большее 40.

Таким образом, в результате работы программы на экран выводится единственное число N — минимальное число, после обработки которого получается число R , большее 40.

Результат работы программы к заданию № 5 представлен на рисунке 6. На экран выводится одно число — 16, которое является ответом к данному заданию.

2.3. Задание № 15. Работа с основными понятиями и законами математической логики

Еще одним заданием, для решения которого можно использовать программу на языке программирования Python, является задание № 15, направленное на проверку знания основных понятий и законов математической логики. В задании дано утверждение: «натуральное число n делится без остатка на натуральное число m » — ДЕЛ(n, m), а также логическая формула:

$$(\text{ДЕЛ}(x, 2) \rightarrow \neg \text{ДЕЛ}(x, 3)) \vee (x + A \geq 70),$$

для которой необходимо найти наименьшее натуральное число A , при котором она будет тождественно истинна при любом натуральном значении переменной x . Соответствующее задание из открытого банка заданий ЕГЭ по информатике представлено на рисунке 7.

Для решения задания можно написать программу на Python, которая будет перебирать натуральные числа A от 1 до 70 и для каждого из этих чисел также будет перебирать натуральные числа x от 1 до 70 и вы-

Обозначим через ДЕЛ(n, m) утверждение «натуральное число n делится без остатка на натуральное число m ».

Для какого **наименьшего** натурального числа A формула

$$(\text{ДЕЛ}(x, 2) \rightarrow \neg \text{ДЕЛ}(x, 3)) \vee (x + A \geq 70)$$

тождественно истинна (т.е. принимает значение 1) при любом натуральном значении переменной x ?

Рис. 7. Задание № 15 из открытого банка заданий ЕГЭ

числять значение логической формулы для каждой из пар чисел A и x (поскольку в логической формуле в одной из скобок присутствует выражение $x + A \geq 70$, которое всегда будет истинным, если $x \geq 70$ или $A \geq 70$). Если при заданном A для каждого числа x логическая формула будет истинной, то первое полученное A — это искомое число.

Программа к заданию № 15 из демонстрационного варианта КИМ ЕГЭ по информатике представлена на рисунке 8.

Данная программа реализует следующий алгоритм:

1) Определяем функцию для утверждения «натуральное число n делится без остатка на натуральное число m » или $\text{ДЕЛ}(n, m)$, которое используется в логической формуле. Для этого в программе определяется функция $\text{delit}(n, m)$, которая возвращает значение True (истина), если выполняется условие «число n делится на число m без остатка» ($n \% m == 0$), в противном случае функция возвращает False (ложь).

2) В основной программе перебираем натуральные числа A от 1 до 70, используя цикл с параметром, и определяем логическую переменную $\text{correct} = \text{True}$, т. е. предполагаем, что текущее число A является искомым

числом. Следующие команды также будем выполнять внутри данного цикла.

3) Перебираем натуральные числа x от 1 до 70, используя цикл с параметром, вычисляем значение логической формулы

$$F = (\text{ДЕЛ}(x, 2) \rightarrow \neg \text{ДЕЛ}(x, 3)) \vee (x + A \geq 70) = \\ = \neg(\text{ДЕЛ}(x, 2) \vee \neg \text{ДЕЛ}(x, 3)) \vee (x + A \geq 70),$$

а затем проверяем, является ли логическая формула ложной, т. е. принимает ли она значение 0. Если данное условие выполняется, то текущее число A не является искомым, т. е. переменная $\text{correct} = \text{False}$.

4) Проверяем, является ли текущее число A искомым числом, т. е. проверяем условие $\text{correct} == \text{True}$. Если данное условие выполняется, на экран выводится искомое число A . Поскольку текущее число A — это минимальное искомое число, при котором логическая формула является истинной при любом натуральном значении переменной x , необходимо завершить перебор чисел (выйти из цикла) при помощи команды break . Если этого не сделать, на экран выведутся все числа A от 1 до 70, для которых логическая функция является истинной при любом натуральном значении переменной x .

```

№15.py - D:\№15.py (3.11.1)
File Edit Format Run Options Window Help
# Определение функции ДЕЛ(n, m), т.е. утверждения «натуральное
# число n делится без остатка на натуральное число m»
def delit(n, m):
    # Проверяем, делится ли число n на число m без остатка
    if n % m == 0:
        # Если условие выполняется - возвращаем истину
        return True
    else:
        # Если условие не выполняется - возвращаем ложь
        return False

# Перебор натуральных чисел A от 1 до 70
for A in range(1, 71):

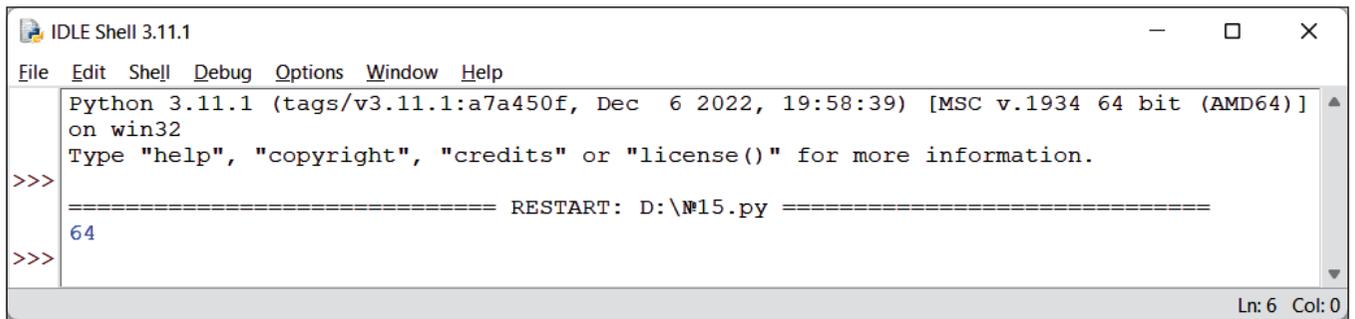
    # Определение логической переменной correct
    # Предположим, что текущее A - искомое число
    correct = True

    # Перебор натуральных чисел x от 1 до 70
    for x in range(1, 71):
        # Вычисление значения логической функции F
        F = (not delit(x, 2) or not delit(x, 3)) or (x + A >= 70)
        # Проверяем, является ли логическая функция ложной
        if F == 0:
            # Если функция принимает значение 0, т.е является
            # ложной, то текущее A не является искомым числом
            correct = False

    # Проверяем, является ли текущее число A искомым
    if correct == True:
        # Выводим на экран исходное число A
        print(A)
        # Поскольку первое выведенное число A будет минимальным числом,
        # выходим из цикла, чтобы не появлялись другие числа
        break
Ln: 1 Col: 0

```

Рис. 8. Программа к заданию № 15 из открытого банка заданий ЕГЭ



```

Python 3.11.1 (tags/v3.11.1:a7a450f, Dec 6 2022, 19:58:39) [MSC v.1934 64 bit (AMD64)]
on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:\№15.py =====
64
>>>

```

Рис. 9. Результат работы программы к заданию № 15 ЕГЭ по информатике

Примечание. При работе с импликацией и эквивалентностью в рамках вычисления значения логической формулы F можно использовать как основные логические операции в Python (инверсию, дизъюнкцию и конъюнкцию), так и операции отношения («меньше или равно» и «равно»). При этом выражение с использованием операций отношения следует заключать в скобки во избежание ошибок при вычислении значения логической формулы.

Таким образом, в результате работы программы на экран выводится минимальное искомое число A , при котором логическая формула является истинной при любом натуральном значении переменной x . Результат работы программы к заданию № 15 представлен на рисунке 9. На экран выводится единственное натуральное число — 64, которое является ответом к заданию.

2.4. Задание № 16. Вычисление рекуррентных выражений

Используя программу на языке программирования Python, также можно решить задание № 16, которое

направлено на проверку умения вычислять рекуррентные выражения. В соответствующем задании демонстрационного варианта КИМ, которое представлено на рисунке 10, необходимо найти значение выражения $F(2023)/F(2020)$ для заданного алгоритма вычисления значения функции $F(n)$.

Для решения этого задания можно написать программу на Python, которая определяет алгоритм вычисления значения функции $F(n)$, и вывести на экран значение выражения $F(2023)/F(2020)$.

Программа к заданию № 16 демонстрационного варианта КИМ ЕГЭ по информатике представлена на рисунке 11. Программа определяет функцию для вычисления значения $F(n)$: если $n = 1$, то функция возвращает значение 1, в противном случае, если $n > 1$, то функция возвращает значение $n \cdot F(n - 1)$. В свою очередь, в основной программе на экран выводится значение выражения $F(2023)/F(2020)$.

Однако данная программа не работает и вместо результата выдает ошибку «RecursionError: maximum recursion depth exceeded». Эта ошибка означает, что при

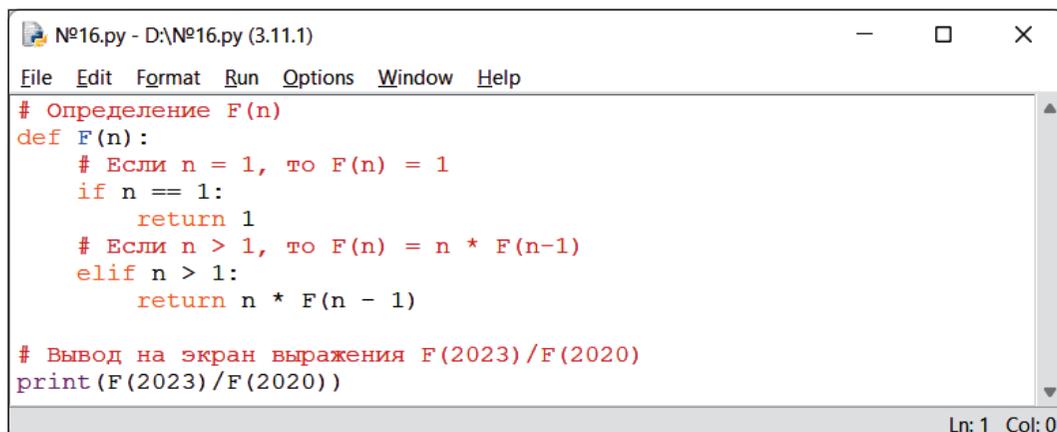
Алгоритм вычисления значения функции $F(n)$, где n – натуральное число, задан следующими соотношениями:

$$F(n) = 1 \text{ при } n = 1;$$

$$F(n) = n \times F(n - 1), \text{ если } n > 1.$$

Чему равно значение выражения $F(2023) / F(2020)$?

Рис. 10. Задание № 16 из демонстрационного варианта КИМ ЕГЭ по информатике



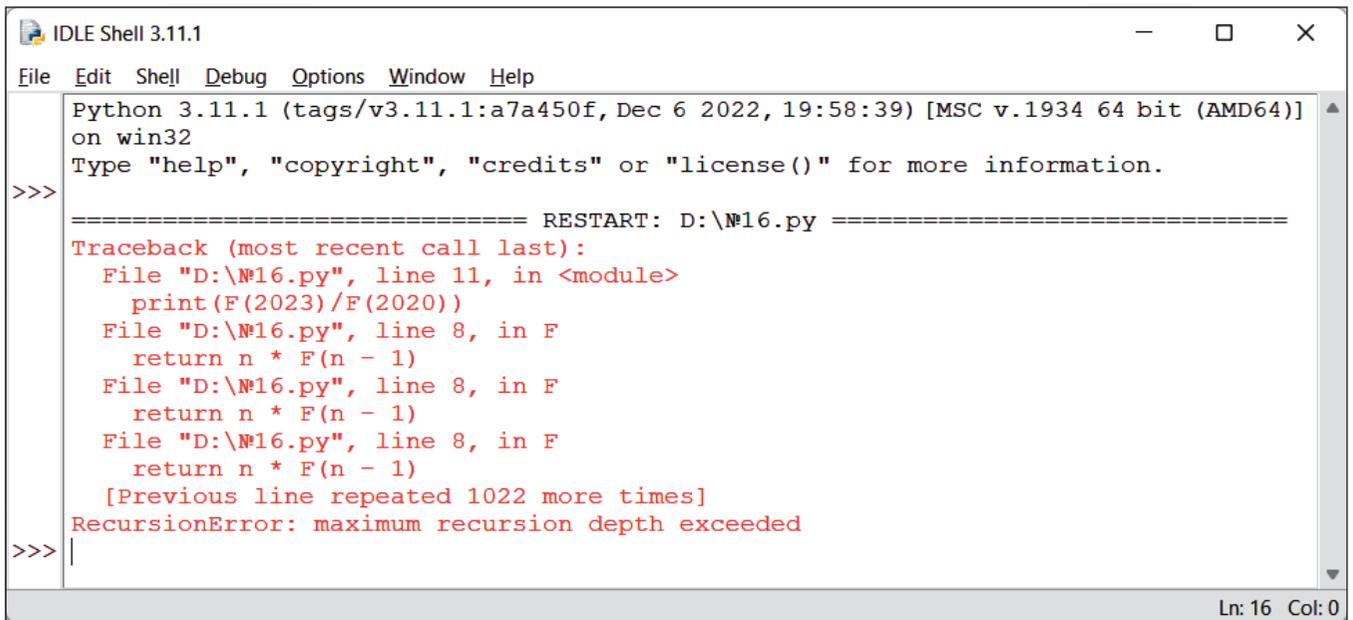
```

№16.py - D:\№16.py (3.11.1)
File Edit Format Run Options Window Help
# Определение F(n)
def F(n):
    # Если n = 1, то F(n) = 1
    if n == 1:
        return 1
    # Если n > 1, то F(n) = n * F(n-1)
    elif n > 1:
        return n * F(n - 1)

# Вывод на экран выражения F(2023)/F(2020)
print(F(2023)/F(2020))
Ln: 1 Col: 0

```

Рис. 11. Первая версия программы к заданию № 16 ЕГЭ по информатике



```

Python 3.11.1 (tags/v3.11.1:a7a450f, Dec 6 2022, 19:58:39) [MSC v.1934 64 bit (AMD64)]
on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:\№16.py =====
Traceback (most recent call last):
  File "D:\№16.py", line 11, in <module>
    print(F(2023)/F(2020))
  File "D:\№16.py", line 8, in F
    return n * F(n - 1)
  File "D:\№16.py", line 8, in F
    return n * F(n - 1)
  File "D:\№16.py", line 8, in F
    return n * F(n - 1)
  [Previous line repeated 1022 more times]
RecursionError: maximum recursion depth exceeded
>>>
Ln: 16 Col: 0

```

Рис. 12. Результат работы первой версии программы к заданию № 16 ЕГЭ по информатике с ошибкой превышения предела рекурсии

выполнении данной программы был превышен предел рекурсии, т. е. функция $F(n)$ вызвала саму себя более 1000 раз (поскольку для нахождения значения каждого из выражений $F(2023)$ и $F(2020)$ функция $F(n)$ должна вызвать себя более 2000 раз). Данное ограничение было установлено в Python для того, чтобы предотвратить бесконечную рекурсию и сбой в работе Python, возникающие при появлении бесконечной рекурсии. Результат работы первой версии программы представлен на рисунке 12.

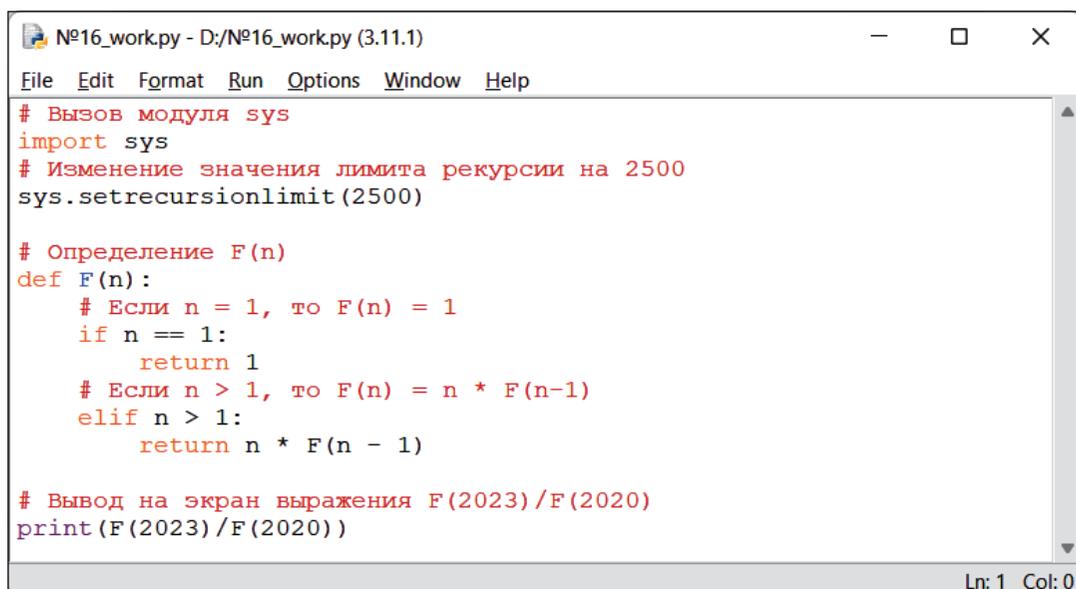
Чтобы программа заработала правильно и выводила на экран значение выражения $F(2023)/F(2020)$, необходимо изменить предел рекурсии в Python. Для этого в начало программы необходимо добавить две команды. Первая команда «import sys» позволяет вызвать модуль

sys (файл с командами, которые позволяют изменить настройки Python). В свою очередь, вторая команда «sys.setrecursionlimit(2500)» дает возможность при помощи метода `setrecursionlimit` изменить предел рекурсии до 2500, т. е. теперь функция $F(n)$ сможет вызывать саму себя при выполнении программы не более 2500 раз.

Исправленная программа к заданию № 16 демонстрационного варианта КИМ ЕГЭ по информатике представлена на рисунке 13.

Результат работы программы представлен на рисунке 14: на экран выводится значение выражения $F(2023)/F(2020)$, это число 8266912626.0, поскольку результатом операции «деление» в языке программирования Python всегда является вещественное число.

Следовательно, ответ к заданию: 8266912626.



```

№16_work.py - D:\№16_work.py (3.11.1)
File Edit Format Run Options Window Help
# Вызов модуля sys
import sys
# Изменение значения лимита рекурсии на 2500
sys.setrecursionlimit(2500)

# Определение F(n)
def F(n):
    # Если n = 1, то F(n) = 1
    if n == 1:
        return 1
    # Если n > 1, то F(n) = n * F(n-1)
    elif n > 1:
        return n * F(n - 1)

# Вывод на экран выражения F(2023)/F(2020)
print(F(2023)/F(2020))
Ln: 1 Col: 0

```

Рис. 13. Исправленная версия программы к заданию № 16 ЕГЭ по информатике

```

IDLE Shell 3.11.1
File Edit Shell Debug Options Window Help
Python 3.11.1 (tags/v3.11.1:a7a450f, Dec 6 2022, 19:58:39) [MSC v.1934 64 bit (AMD64)]
on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:/№16_work.py =====
8266912626.0
>>>
Ln: 6 Col: 0

```

Рис. 14. Результат работы программы к заданию №16 ЕГЭ по информатике

3. Заключение

Таким образом, возможности языка программирования Python могут использоваться для решения не только задач раздела «Программирование», но и других заданий ЕГЭ по информатике. Однако для использования Python в качестве альтернативного инструмента для решения задач, обучающимся необходимо знать не только особенности реализации основных алгоритмических структур (следование, ветвление, повторение), но и основные встроенные функции языка, а также уметь работать со вспомогательными алгоритмами (процедурами, функциями) и строками.

Разумеется, данный способ решения предложенных задач «в лоб», фактически полным перебором возможных ответов, может вызывать сложные чувства у учителей информатики, поскольку ученик минует этап логических размышлений и выводов, поиска оптимального пути решения. Однако этот способ учит учеников формализовать постановку задачи на языке, понятном компьютеру, что тоже является одним из результатов обучения по информатике. Кроме того, он требует наличия некоторых знаний о встроенных функциях, выходящих за пределы базового школьного курса программирования. А в условиях ограниченного времени на экзамене обучающиеся, хорошо владеющие языком Python, должны уметь использовать те возможности, которые дает «программный» метод решения задач, напрямую не относящихся к программированию.

Список источников

1. Ведомости: «Рособрнадзор сообщил о росте интереса школьников к сдаче ЕГЭ по информатике» // Федеральная служба по надзору в сфере образования и науки. 26 мая 2022 г. <https://obrnadzor.gov.ru/smi-o-rosobrnadzore/vedomosti-rosobrnadzor-soobshhil-o-roste-interesa-shkolnikov-k-sdache-ege-po-informatike/>
2. Лутц М. Изучаем Python: пер. с англ. Ю. Н. Артеменко. 5-е изд. СПб.: Диалектика, 2019. 832 с.
3. Маркелов В. К., Завьялова О. А. Возможности языка Python 3.10 при обучении программированию в школе // Шуй-

ская сессия студентов, аспирантов, педагогов, молодых ученых. Материалы XIV Международной научной конференции, посвященной Году науки и технологий Российской Федерации, 205-летию начала подготовки педагогов в Ивановской области (Москва — Иваново — Шуя, 06–07 октября 2021 года) / отв. редактор А. А. Червова. М.-Иваново-Шуя: Шуйский филиал федерального государственного бюджетного образовательного учреждения высшего образования «Ивановский государственный университет», 2021. С. 96–97. EDN: XXZKLZ.

4. Маркелов В. К., Завьялова О. А. Инструментарий облачной среды Google Colaboratory для обучения программированию и подготовки к ЕГЭ по информатике // Информатика в школе. 2022. № 6. С. 68–80. EDN: KUDDFH. DOI: 10.32517/2221-1993-2022-21-6-68-80

5. Панова И. В., Коливык А. А. Методические аспекты обучения программированию на языке Python в школьном курсе информатики // Информатика в школе. 2020. № 6. С. 47–50. EDN: IMYAAI. DOI: 10.32517/2221-1993-2020-19-6-47-50

6. Пирогов В. Ю. Особенности преподавания программирования на языке Python // Современное образование: традиции и инновации. Материалы международной научно-практической конференции. В 2 ч. Ч. 1 (Шадринск, 25 октября 2019 года). Шадринск: Шадринский государственный педагогический университет, 2020. С. 225–226. EDN: BIWDRW.

7. Письмо Рособрнадзора от 28.01.2021 № 10-21 «О направлении разъяснений по отдельным вопросам, связанным с организацией работы конфликтных и предметных комиссий по учебному предмету “Информатика и информационно-коммуникационные технологии (ИКТ)”». <https://legalacts.ru/doc/pismo-rosobrnadzora-ot-28012021-n-10-21-o-napravlenii-razjasnenii/>

8. Письмо Рособрнадзора от 31.01.2022 № 04-18 «О направлении методических документов, рекомендуемых при организации и проведении государственной итоговой аттестации по образовательным программам основного общего образования и среднего общего образования в 2022 году». https://www.consultant.ru/document/cons_doc_LAW_408305/

9. Приказ Минпросвещения России от 31.05.2021 № 287 «Об утверждении федерального государственного образовательного стандарта основного общего образования». https://www.consultant.ru/document/cons_doc_LAW_389560/

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