

УДК 37.04:372.8
DOI 10.17513/snt.40101

ОБУЧЕНИЕ АНАЛИЗУ ЗАПИСИ ЧИСЕЛ И ПРОВЕРКЕ ИХ ДЕЛИМОСТИ В ЯЗЫКЕ ПРОГРАММИРОВАНИЯ PYTHON В ШКОЛЕ

¹Козлов С.В., ²Быков А.А.

¹ФГБОУ ВО «Смоленский государственный университет», Смоленск,
e-mail: svkozlov1981@yandex.ru;

²ФГБОУ ВО «Национальный исследовательский университет «МЭИ»,
Смоленский филиал, Смоленск, e-mail: alex1by@mail.ru

Приложения анализа значений числовых выражений и проверки их делимости играют важную роль при исследовании многих практических ситуаций. Прикладной характер этого вопроса обуславливает поиск оптимальных путей их решения. В связи с этим использование инструментов языков программирования приобретает особую значимость. Обучение решению таких задач оптимальным образом, наряду с изучением базовых алгоритмических конструкций, образует ядро линии программирования в школьном курсе информатики. Ввиду этого в статье обсуждаются возможности применения средств языка программирования Python для решения таких задач. Основное внимание при составлении алгоритмов анализа чисел из заданного отрезка на соответствие указанному шаблону и проверке их на делимость уделяется вопросу оптимальности. Авторами сравниваются особенности переборного и оптимального вариантов решения задачи. Переборный вариант решения сначала предусматривает проверку всех чисел на соответствие маске. После чего выбранные числовые значения проверяются на делимость. Оптимальный подход основывается на проверке маски только чисел, которые делятся на указанное в задании число. В связи с чем в статье демонстрируется принцип программной реализации, при которой задается шаг перехода от одного числа, удовлетворяющего критерию делимости, к другому числу. В ходе исследовательской работы, представленной авторами, показана необходимость овладения навыками организации оптимальной обработки числовых данных при обучении информатике в школе на профильном уровне.

Ключевые слова: информатика, программирование, алгоритм, арифметическое выражение, числа, делимость, язык Python, компьютерная программа, образовательный процесс

LEARNING TO ANALYZE NUMBER RECORDING AND CHECK THEIR DIVISIBILITY IN THE PYTHON PROGRAMMING LANGUAGE AT SCHOOL

¹Kozlov S.V., ²Bykov A.A.

¹Smolensk State University, Smolensk, e-mail: svkozlov1981@yandex.ru;

²National Research University MPEI, Smolensk branch, Smolensk, e-mail: alex1by@mail.ru

Applications of analyzing the values of numerical expressions and testing their divisibility play an important role in the study of many practical situations. The applied nature of this issue determines the search for optimal ways to solve them. In this regard, the use of programming language tools becomes especially important. Teaching to solve such problems in an optimal way, along with studying basic algorithmic structures, forms the core of the programming line in the school computer science course. In view of this, the article discusses the possibilities of using the Python programming language to solve such problems. When developing algorithms for analyzing numbers from a given segment for compliance with a specified pattern and testing them for divisibility, the main attention is paid to the issue of optimality. The authors compare the features of exhaustive and optimal solutions to the problem. The exhaustive version of the solution first involves checking all numbers for compliance with the mask. After which the selected numerical values are checked for divisibility. The optimal approach is based on checking the mask only for numbers that are divisible by the number specified in the task. In this connection, the article demonstrates the principle of software implementation, in which the step of transition from one number that satisfies the divisibility criterion to another number is specified. In the course of the research work presented by the authors, the need to master the skills of organizing optimal processing of numerical data when teaching computer science at school at the profile level is shown.

Keywords: computer science, programming, algorithm, arithmetic expression, numbers, divisibility, Python language, computer program, educational process

Введение

Тема «Системы счисления» является сквозной при изучении школьного курса информатики [1]. Знания и умения по этой теме выступают не только предметом изучения, они используются при решении многих заданий из смежных тем [2]. Напри-

мер, при изучении способов кодирования и декодирования данных, при анализе вычислительных алгоритмов для формальных исполнителей, при составлении таблиц истинности для логических функций, при обработке IP-адресов в компьютерных сетях. При этом в профильном курсе информатики

в старших классах особое внимание уделяется анализу операндов арифметических выражений [3]. В частности, обучающиеся изучают вопрос о значении цифр в записи числовых выражений, выборе чисел из указанного диапазона, удовлетворяющих заданному шаблону и критериям делимости. Для этого им необходима фундаментальная подготовка в области теоретических знаний из математики и получение навыков решения практических заданий. Однако заметим, что во многих случаях в школе ограничиваются только разбором теоретических подходов и решением некоторого набора задач «вручную».

Привлечение к решению таких задач инструментальных средств современных систем программирования [4] существенным образом смещает акценты при анализе чисел в различных системах счисления. Ученик получает действенный инструмент приложения своих теоретических знаний. Он «видит» результаты работы алгоритмов обработки числовых значений и получает их эффективным образом, избегая громоздких вычислений на бумаге. В то же время проверка умения выполнять задания на анализ чисел, удовлетворяющих заданным условиям отбора, на ЕГЭ по информатике говорит о том, что обучающиеся в недостаточной степени овладевают навыками их решения. В связи с этим необходимо отводить в системе обучения информатике в школе на профильном уровне при решении таких заданий с использованием систем программирования, например таких как Python, больше времени.

Цель исследования – оценка эффективности объяснения обучающимся методов анализа записи чисел и проверки их делимости в языке программирования Python.

Научная новизна заключается в апробации методологии внедрения функциональных инструментов языка программирования Python среды PyCharm при решении задач выборки чисел и анализа полученных значений, которые удовлетворяют заданным критериям делимости.

Материалы и методы исследования

Начало изучения среды программирования Python относится к средней школе [5, 6]. В старших классах обучающиеся профильных классов учатся применять изученные алгоритмические конструкции и базовые алгоритмы на практике решения разных прикладных заданий. Так, например, многие из этих заданий сводятся к анализу числовых значений в указанном диапазоне [7, 8], а затем последующему выбору из них числового выражения по представленному шаблону и проверке полученного результата на делимость числа n на m . В ЕГЭ по информатике в таком задании диапазон указан сразу, при этом его границы сознательно расширяют так, чтобы он содержал числа с восемью и более цифрами. Это усложняет не только проводимый в задании анализ, но и предполагает написание для решения оптимального эффективного программного кода, а не использование в нем переборного варианта. В связи с этим обучающиеся должны понимать, что в таком случае лучше «отталкиваться» от делимости чисел. Для этого необходимо для организации цикла сначала вычислить начальное значение из заданного диапазона, которое удовлетворяет критерию делимости. Затем с шагом, равным делителю, проверять в цикле числа на соответствие указанному шаблону. Эти действия следует повторять до тех пор, пока не будет достигнут правый конец диапазона.

Обобщим подходы к решению таких примеров и проанализируем способы их решения.

Задача. Сколько среди чисел из отрезка [12436578; 45653317] удовлетворяют маске $3?3215*7$ и делятся на 24 без остатка? В ответе укажите все найденные числа в порядке возрастания и результат их деления на 24 в виде пар значений.

Представим сначала решение данной задачи на языке программирования Python переборным способом:

```
for i in range(12436578, 45653317 + 1):
    if str(i)[0] == '3' and str(i)[2:6] == '3215' and str(i)[-1] == '0':
        if i % 24 == 0:
            print(i, i // 24)
```

Искомым ответом будет следующий набор пар чисел:

```
31321560 1305065
32321520 1346730
34321560 1430065
35321520 1471730
37321560 1555065
38321520 1596730
```

В данном решении для организации проверки чисел на соответствие шаблону и критерию делимости использован цикл с параметром `for`. Обратим внимание, что для определения искомого отрезка проверки чисел в операторе `range` левый конец включается, а правый нет. Поэтому к правому числу полуинтервала необходимо прибавить единицу.

В цикле с помощью первого оператора `if` анализируется, подходит ли текущее число под заданный шаблон. Для этого обучающиеся должны владеть навыками работы со срезами списков. Проверяемое число переводится в строковый формат, после чего становятся доступны операции срезов. Так как нумерация в языке Python начинается с нуля, то проверка первого символа шаблона осуществляется с помощью команды `str(i)[0] == '3'`. Второй символ может быть любым, поэтому его проверка не требуется. С третьего по шестое место в числе должны стоять символы '3215'. Их проверка осуществляется с помощью среза `str(i)[2:6]`, так как, напомним, в Python диапазон представляет собой полуинтервал, левый конец которого включен в него, а правый – нет. Вместо седьмого символа в шаблоне указана '*', а не как на втором месте '?', в связи с этим на ее месте может быть последовательность символов любой длины, в том числе

и нулевой. Поэтому для того, чтобы указать, что последний символ строки – это цифра '0', необходимо воспользоваться двойной нумерацией позиций в списках языка программирования Python. Так, при нумерации элементов списка с конца, справа налево, они имеют индексы, начиная с -1 в обратную сторону по порядку. То есть последний символ строки будет иметь индекс -1, несмотря на то, находится ли он на седьмой, восьмой или иной позиции в числе.

С помощью второго оператора `if` осуществляется проверка на делимость на 24. Для этого вычисляется остаток от деления числа, удовлетворяющего заданному шаблону, на 24 и сравнивается с нулем. Если проверка завершена успешно, то искомое число из заданного отрезка найдено, и оно выводится на экран вместе с результатом его целочисленного деления на 24 с помощью команды `print`.

Вывод шести пар найденных значений в рассмотренном варианте для диапазона восьмизначных чисел будет занимать около 5 с. Расширение заданного промежутка еще больше увеличит время выполнения программы, что становится на экзамене критичным. В связи с этим обучающимся необходимо предложить вариант оптимизации представленного программного кода. Рассмотрим его:

```
for i in range(12436578 // 24 + 1, 45653317 // 24 + 1):
    num = str(24 * i)
    if num[0] == '3' and num[2:6] == '3215' and num[-1] == '0':
        print(num, i)
```

В данном программном коде цикл с параметром `for` осуществляется только по числам, которые делятся на 24. Для этого левый и правый конец заданного диапазона восьмизначных чисел делится нацело на 24 и к полученным результатам прибавляется единица. Затем в цикле для каждого из полученных значений формируется строка, которая является результатом их умножения на 24. Таким способом получаем набор чисел в строковом формате с шагом 24, что сокращает количество витков цикла 24 раза. Последующая проверка чисел, которые уже делятся на 24, на соответствие заданному шаблону осуществляется тем же самым способом. Такой подход к решению поставленной задачи позволяет значительно выиграть во времени выполнения программы [9, 10]. В представленном решении результат становится доступным мгновенно в течение нескольких миллисекунд. При этом заметим, что в представленном способе решения обучающиеся, изучая

приемы оптимизации программного кода, параллельно знакомятся с организацией циклов с указанным шагом в соответствии с критериями делимости натуральных чисел [11]. А в контексте ЕГЭ по информатике это позволяет им получить ответ на задание оптимальным способом за минимальный промежуток времени.

Такие задания были предложены обучающимся на формирующем и констатирующем этапах педагогического эксперимента, которые позволили сформировать методологическую базу исследования. В ходе экспериментальной деятельности для обработки полученных результатов были применены математические методы анализа, в частности статистические методы, математический аппарат имплицитивных матриц и соответствие Галуа. Гипотеза исследования заключалась в том, что использование алгоритмических конструкций языка программирования Python, его структур данных и инструментов работы с ними

в школе при анализе записи чисел и проверке их на делимость заданному числу повышает эффективность обучения.

Педагогический эксперимент по обучению разным подходам к анализу записи чисел и проверке их делимости в языке программирования Python проводился в течение четырех недель в двух образовательных учреждениях Смоленской области. А именно, в Смоленском физико-математическом лицее при МИФИ и средней школе № 6 г. Смоленска. Информатика в этих учебных заведениях в физико-математическом классе и IT-классе соответственно изучается на профильном уровне. В экспериментальной работе приняли участие 32 обучающихся.

Результаты исследования и их обсуждение

Результаты педагогического исследования, полученные в ходе диагностических работ, обобщены в табл. 1 и 2.

Констатирующий этап эксперимента состоял в изучении теоретического материала и решении заданий без применения возможностей сред программирования. Учащиеся сначала познакомились с понятием «маска», выясняли отличия между используемыми в ней для отбора произвольных значений символами «?» и «*». Затем они повторяли критерии делимости и исследовали способы обработки чисел в зависимости от заданных промежутков отбора и значений делителей. Решение предложенных заданий преимуще-

ственно осуществлялось переборными способами. В случаях небольшого набора данных обучающиеся решали задачи вручную, а в ситуациях обработки чисел с указанных отрезков соизмеримой длины, как правило, в диапазоне от четырехзначных до семизначных чисел, они применяли инструменты для организации прогрессий, фильтров и вычислений делителей из электронных таблиц. На завершающей стадии этого этапа эксперимента обучающимся был предложен тест, который состоял из шести заданий. Все задания было необходимо решить без использования средств обработки данных в системах программирования, в частности алгоритмических конструкций и функций языка Python, иными словами, без программных инструментов оптимизации.

Формирующий этап эксперимента предполагал для решения таких задач широкое применение языка программирования Python. Сначала обучающиеся тренировались в программной реализации переборных алгоритмов для определения количества чисел, соответствующих заданной маске и делящихся на указанное число без остатка. Затем они учились оптимизировать составленные ими алгоритмы, сравнивали их сложность и время выполнения программы. При этом особое внимание также уделялось эффективности используемых алгоритмических подходов не только по времени, но и по памяти. Это требовало от них расширения круга своих знаний и умений при решении таких задач.

Таблица 1

Результаты констатирующего этапа педагогического эксперимента

Группа	Число обучающихся, достигших уровня усвоения знаний и освоения умений			Всего
	Высокий	Повышенный	Базовый	
СФМЛ при МИФИ	3	6	9	18
IT-класс школа № 6	1	6	7	14
Всего	4	12	16	32

Таблица 2

Результаты формирующего этапа педагогического эксперимента

Группа	Число обучающихся, достигших уровня усвоения знаний и освоения умений			Всего
	Высокий	Повышенный	Базовый	
СФМЛ при МИФИ	8	7	3	18
IT-класс школа № 6	7	5	2	14
Всего	15	12	5	32

На завершающей стадии формирующего этапа эксперимента обучающимся также был предложен диагностический тест. Данный итоговый тест содержал шесть заданий. Все задания было необходимо решить с помощью языка программирования Python. Отметим, что вычислительная сложность заданий была такова, что обработка данных задачи без использования средств оптимизации в алгоритмах было невозможно. Данные, полученные в ходе экспериментальной работы, были обработаны в интеллектуальной программной оболочке «Advanced Tester». При этом программное приложение «Advanced Tester», используемое в эксперименте, позволяло не только анализировать полученные результаты диагностических срезов знаний и умений обучающихся, но и моделировать на всем его протяжении индивидуальные и групповые образовательные карты учащихся с помощью математического аппарата имплицитных матриц и соответствия Галуа [12, 13]. Кроме того, возможности удаленного использования этого программного средства открывали дополнительные горизонты совершенствования своих навыков обучающимися во внеучебное время [14, 15]. Данные о текущем характере обучения фиксировались и обрабатывались в программной среде «Advanced Tester» автоматически.

Качественный анализ условий и результатов эксперимента. Результаты, полученные в ходе исследования и отраженные в таблицах, позволяют утверждать, что уровень знаний и умений обучающихся увеличивается во всех диапазонах измерений. Это указывает на то, что применение инструментов языка программирования, в частности Python в среде PyCharm, упрощает анализ условий отбора данных по шаблону и проверку делимости выражения на заданное натуральное число. Кроме того, существенно расширяется диапазон рассматриваемых в задачах чисел. Применение эффективного алгоритма обеспечивает выполнение программы за доли миллисекунды для больших многозначных чисел, содержащих от десяти до двадцати знаков. Обучающиеся, изучая на углубленном уровне алгоритмы оптимизации отбора чисел по заданным критериям, пишут программный код решения задачи с более сложными условиями. При этом принципы его построения инвариантны относительно исследуемого диапазона числовых выражений. Это обуславливает преимущество использования программной среды языка Python от решения задач по анализу записи чисел и проверке их делимости «вручную» и средствами электронных таблиц.

Также отметим, что использование при проведении эксперимента программной среды «Advanced Tester» позволило более точно подойти к формированию ориентиров из зоны ближайшего развития обучающихся. Автоматизация подбора индивидуальных заданий от простого к сложному в соответствии с неусвоенными в ходе практики элементами знаний отражает качественный переход учащихся в группы с более высокими показателями учебных достижений. Как во время занятий, так и во внеурочное время обучающиеся имели доступ к непрерывной работе в системе «Advanced Tester», что открывало перед ними перспективы дополнительной тренировки при отработке полученных навыков применения алгоритмов в новых условиях и при необходимости коррекции базовых учебных действий. Формирование подвижных малых групп отражало текущий уровень знаний и умений обучающихся и позволяло гибко адаптировать учебный процесс под их индивидуальные и групповые запросы. В совокупности это как создает прочную основу для решения заданий базового уровня сложности, так и формирует потребность в переходе обучающихся к исследованию более сложных ситуационных аспектов. Обучающиеся не боятся изучать углубленный курс информатики, они видят общие закономерности в его построении и успешно осваивают их. Учащиеся выполняют учебный план с опережением, тем самым высвобождая необходимое время на изучение дополнительного профильного материала. Таким образом, формируется фундаментальная база их знаний и умений, как будущего специалиста в сфере цифровых технологий. Это говорит о том, что гипотеза исследования о повышении эффективности обучения анализу записи чисел и проверке их делимости в школе с использованием возможностей языка программирования Python находит свое доказательство.

Заключение

Итак, экспериментальная работа указывает на целесообразность применения инструментов языка программирования Python при решении задач анализа соответствия записи чисел указанной маске и проверке их делимости на заданное число. Осваивая знания по этой теме, обучающиеся на практике могут понять в какой ситуации лучше использовать то или иное программное средство, либо не использовать его вовсе. При этом они видят всестороннее применение методов и средств программирования, как основополагающей учебной единицы школьного курса

профильной информатики. Обучающиеся апробируют на практике применение теоретических подходов при решении разных прикладных задач. При таком подходе к обучению они учатся после проведенного анализа осознанно применять тот или иной программный инструмент, который в данной ситуации обеспечивает эффективность и позволяет получить ответ на поставленный в задаче вопрос в кратчайшие сроки с меньшими функциональными вложениями. Это способствует пониманию всех нюансов применяемого алгоритма, а не только общих принципов его работы. Тем самым у обучающихся происходит формирование профессиональных навыков своей будущей деятельности в области использования цифровых технологий в разных отраслях науки и техники.

Список литературы

1. Латышева Е.В. Тема «Системы счисления» в школьном курсе информатики // Информационные технологии в образовательном процессе вуза и школы. Материалы XV Всероссийской научно-практической конференции / Редакция: Р.М. Чудинский (науч. ред.) и др. Воронеж, 2021. С. 269–273.
2. Аванесян А.А., Разумова И.В., Донов Д.Д. Системы счисления в математике и их применение в жизни // Университетская наука. 2022. № 1 (13). С. 131–136.
3. Иорданский М.А., Мухин Н.А. Использование информационно-педагогических технологий при изучении систем счисления и компьютерной арифметики // Информатика и образование. 2017. № 9 (288). С. 51–53.
4. Козлов С.В., Быков А.А. Обучение школьников выполнению и анализу алгоритмов для формального исполнителя с использованием систем программирования // Современные проблемы науки и образования. 2023. № 2. URL: <https://science-education.ru/ru/article/view?id=32491> (дата обращения: 15.04.2024).
5. Бровка Н.В., Францкевич А.А. Обучение учащихся основам алгоритмизации и программирования // Педагогика информатики. 2020. № 3. С. 1–9.
6. Козлов С.В. Особенности обучения школьников информатике в профильной школе // Концепт. 2014. № 1. С. 31–35. URL: <http://e-koncept.ru/2014/14006.htm> (дата обращения: 17.05.2024).
7. Архангельская Е.В. Организация обучения основам алгоритмизации и программирования с использованием анализа математических задач // Информатизация образования и науки. 2022. № 4 (56). С. 166–175.
8. Каган Э.М. Применение визуальных языков программирования для повышения эффективности обучения разделу «Алгоритмизация и программирование» школьного курса информатики // Вестник Московского городского педагогического университета. Серия: Информатика и информатизация образования. 2018. № 1 (43). С. 99–104.
9. Пономарев Д.А. Решение олимпиадных задач по информатике по теме «Системы счисления» на языке программирования Python // Информационные технологии в образовании. 2021. № 4. С. 193–198.
10. Ильченко О.Ю., Сырицына В.Н., Кадеева О.Е. Решение задач ЕГЭ по информатике средствами языка Python // Высшее образование сегодня. 2021. № 11–12. С. 42–54.
11. Рослякова Е.А., Химич А.М. Преимущества использования языка программирования Python при изучении раздела «Алгоритмизация и программирование» в школьном курсе информатики // Современные тенденции развития фундаментальных и прикладных наук: материалы Всероссийской с международным участием научно-практической конференции / Под ред. С.А. Коньшаковой. 2018. С. 148–152.
12. Козлов С.В., Быков А.А. Особенности изучения междисциплинарных тем школьных курсов математики и информатики с помощью методов математического моделирования // Проблемы современного образования. 2021. № 5. С. 250–261.
13. Козлов С.В., Быков А.А. Применение методов математического моделирования для диагностики знаний школьников // Современные наукоемкие технологии. 2021. № 4. С. 157–162.
14. Киселева О.М. Программные средства поддержки удаленного обучения // Вызовы цифровой экономики: тренды развития в условиях последствий пандемии COVID-19: сборник статей IV Всероссийской научно-практической конференции, приуроченной к Году науки и технологий в России. Брянск, 2021. С. 143–146.
15. Senkina G.E., Timofeeva N.M., Kiseleva O.M. Modernization of traditional educational forms in the context of distance learning // Journal of Higher Education Theory and Practice. 2022. Т. 22, № 3. P. 160–165.