

Министерство образования Кировской области

Кировское областное государственное образовательное автономное
учреждение дополнительного профессионального образования
«Институт развития образования Кировской области»

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ
по решению сложных задач
с использованием среды
программирования Python

Киров 2025

УДК 377
ББК 74.4
М 54

*Печатается по решению Совета по научной,
инновационной и редакционно-издательской деятельности
КОГОАУ ДПО «ИРО Кировской области»*

Автор-составитель:

Ярославцев В.Л., методист центра непрерывного повышения профессионального мастерства педагогических работников КОГОАУ ДПО «ИРО Кировской области»

Рецензенты:

Кобелева Г.А., директор центра непрерывного повышения профессионального мастерства педагогических работников КОГОАУ ДПО «ИРО Кировской области»

Солкин М.С., учитель информатики КОГОБУ «Лицей №9» г. Слободского

М54 Методические рекомендации по решению сложных задач с использованием среды программирования Python / В.Л. Ярославцев; КОГОАУ ДПО «ИРО Кировской области». – Киров: Полиграфовна, 2025. – 76 с.

Методические рекомендации являются результатом деятельности творческой лаборатории по теме «Методика преподавания языка программирования Python на уроках информатики и во внеурочной деятельности в основной школе». В сборнике представлена методика работы с обучающимися по изучению среды программирования Python, приведено большое количество примеров заданий, в электронных приложениях представлены листинги программ, а также все задания, их решения и методические рекомендации по решению практической части раздела «Язык программирования» учебников Л.Л. Босовой «Информатика» (8–9 классы).

Данное издание может представлять интерес для учителей и преподавателей информатики общеобразовательных и профессиональных образовательных организаций, педагогов дополнительного образования.

© В.Л. Ярославцев, 2025
© ИРО Кировской области, 2025

ОГЛАВЛЕНИЕ

Введение	4
Основы использования среды программирования Python на уроках информатики	6
Базовые понятия и действия	7
Переменные	7
Математические операции	7
Ввод и вывод данных	8
Реализация алгоритмических конструкций	10
Цикл	10
Условие	11
Работа с массивами.....	12
Особенности выполнения задания №16 ОГЭ по информатике	18
Разбор типовых заданий	18
Задача 1. Определение количества чисел в последовательности.....	18
Задача 2. Определение количества чисел в последовательности, оканчивающихся на N	20
Задача 3. Определение суммы элементов последовательности с учетом сложного условия	22
Задача 4. Определение суммы чисел в заданном интервале с учетом сложного условия	24
Задача 5. Определение максимального числа в последовательности с учетом заданного условия	25
Задача 6. Определение минимального числа в последовательности с учетом заданного условия	27
Задача 7. Учет признака окончания последовательности.....	28
Сборник задач с решениями из открытого банка тестовых заданий с сайта ФГБНУ «Федеральный институт педагогических измерений»	31
Приложение	74
Электронное приложение 1. Листинги программ, представленных в разделе «Сборник задач с решениями из открытого банка тестовых заданий с сайта ФГБНУ «Федеральный институт педагогических измерений».....	74
Электронное приложение 2. Реализация практической части раздела «Язык программирования» учебников Л.Л. Босовой «Информатика» (8–9 классы)	74

ВВЕДЕНИЕ

Современное образование требует от учителей не только передачи знаний, но и развития у школьников навыков, актуальных в условиях цифровизации. Программирование занимает особое место в этом процессе, а язык Python, благодаря своей лаконичности и универсальности, становится надёжным помощником в освоении алгоритмического мышления.

Язык Python выбран неслучайно. Его простой синтаксис позволяет ученикам сосредоточиться на логике, а не на сложностях языка, что особенно важно при выполнении заданий ОГЭ, где время ограничено, а задачи требуют чёткого алгоритмического подхода. Кроме того, Python широко применяется за пределами школы – в анализе данных, веб-разработке, машинном обучении, что мотивирует школьников видеть в программировании не только экзаменационную задачу, но и перспективу для дальнейшего развития.

Настоящие методические рекомендации созданы для поддержки педагогов при подготовке обучающихся 8-9 классов к выполнению задания №16 ОГЭ по информатике. В данном задании проверяется умение работать с алгоритмами.

Материалы объединяют теоретические основы, практические задания и методические советы, направленные на эффективное преподавание Python как в рамках уроков, так и во внеурочной деятельности.

Основная цель рекомендаций – познакомить учителей с возможностями Python, необходимыми в том числе для решения экзаменационных задач, и предложить готовые инструменты для интеграции языка в учебный процесс. В материалах разобраны вопросы, связанные с типами переменных, применением математических операций, алгоритмических конструкций, работой с массивами. Каждая тема сопровождается наглядными примерами кода с комментариями, что помогает ученикам визуализировать логику программ. Например, при объяснении циклов демонстрируется разница между `for` и `while`, а в разделе об условиях – использование операторов `if-elif-else` для создания разветвлённых алгоритмов.

Особое внимание уделено практической части. В сборник включены разные типовые задачи, включая поиск минимальных и максимальных значений, подсчёт элементов по условиям, работу с системами счисления. Каждое задание сопровождается пошаговым решением, пояснениями и советами по проверке. Особое внимание уделяется решению упражнений из открытого банка заданий ФИПИ, приведены в электронном виде листинги программ.

Данные рекомендации – результат обобщения опыта педагогов Кировской области, апробировавших материалы на практике. В разработке материалов приняли участие:

Боброва Наталья Станиславовна, учитель информатики МОАУ «Лицей №21» города Кирова;

Булатова Марина Андреевна, учитель информатики МБОУ «ЛГ» г. Кирова;

Загоскина Татьяна Валерьевна, учитель информатики и математики МБОУ СОШ №18 г. Кирова;

Карачёва Ирина Владимировна, учитель математики и информатики МБОУ СОШ №22 г. Кирова;

Клабукова Виктория Валерьевна, учитель информатики МОАУ СОШ с УИОП №37 г. Кирова;

Копытов Николай Анатольевич, директор, учитель информатики МКОУ СОШ п. Октябрьский Фаленского района;

Криницына Екатерина Александровна, учитель информатики МОАУ «Лицей №21» города Кирова;

Ляпустина Ольга Сергеевна, учитель информатики и физики МКОУ СОШ с. Лазарево Уржумского района;

Наумец Светлана Геннадьевна, учитель информатики МКОУ СОШ №3 г. Уржума;

Носкова Надежда Викторовна, учитель информатики КОГОбУ СШ с УИОП пгт Афанасьево;

Овсянников Алексей Леонидович, учитель математики и информатики МКОУ ООШ с. Верхосунье Фаленского района;

Рычкова Наталия Викторовна, заместитель директора по УВР, учитель информатики МОАУ СОШ с УИОП №37 г. Кирова;

Чиркова Анна Васильевна, учитель информатики МБОУ «Лицей» г. Кирово-Чепецка;

Шамянова Анжелика Маратовна, учитель математики МБОУ СОШ №20 г. Кирова;

Шешина Юлия Олеговна, учитель информатики КОГОбУ СШ пгт. Вахруши Слободского района;

Шихалеева Татьяна Леонидовна, учитель информатики МБОУ СОШ №4 пгт Песковка;

Эсаулова Наталья Владимировна, заместитель директора по УВР, учитель информатики КОГОбУ СШ пгт Оричи;

Ярославцев Виктор Леонидович, методист ЦНППМ ПР КОГОАУ ДПО «ИРО Кировской области».

ОСНОВЫ ИСПОЛЬЗОВАНИЯ СРЕДЫ ПРОГРАММИРОВАНИЯ PYTHON НА УРОКАХ ИНФОРМАТИКИ

Python является одним из самых популярных языков программирования для решения задач ОГЭ по информатике. Его простота и лаконичность синтаксиса делают его удобным как для начинающих, так и для опытных программистов. Он обладает множеством встроенных функций и библиотек, что обеспечивает широкий спектр возможностей. Использование функции *input* позволяет пользователю вводить данные в программу.

Python поддерживает операции на числах и строках, упрощая арифметические и строковые операции. Данный язык программирования позволяет работать со списками и массивами, что часто используется при решении задач ОГЭ по информатике. Списки могут содержать различные типы данных и обеспечивают простой доступ к элементам с помощью индексов. Python предлагает удобные методы для модификации, сортировки и поиска элементов в списке.

В Python на высоком уровне представлена поддержка условных операторов *if-elif-else*, что полезно при принятии различных решений в зависимости от значений переменных. Функции позволяют разбивать программу на небольшие логические блоки, облегчая чтение и отладку кода. Модули обеспечивают организацию кода в отдельные файлы для лучшей структурированности проекта.

Python обладает мощной системой обработки исключений, что помогает обрабатывать возможные ошибки в программе и предотвращать ее прерывание. Это позволяет оптимизировать код и улучшить его надежность.

В целом, использование Python для решения задач ОГЭ по информатике является эффективным и удобным выбором благодаря его простоте, мощности и возможностям. Python позволяет программистам легко и эффективно решать широкий спектр задач, что делает его идеальным инструментом для подготовки к ОГЭ по информатике при выполнении задания №16.

В этом задании по выбору применяется навык программирования и задействована тема «Алгоритмизация и программирование» учебного предмета «Информатика» (8-9 классы). Для успешного решения задания потребуются знания таких вещей из программирования как «*переменные*», «*циклы*», «*математические операции*», «*условия*» и «*массивы*». Важно понимание, как эти модули работают отдельно друг от друга и как можно их совмещать. Без массивов можно сделать все задания, но, зная их, можно более легко подойти к решению.

Рассмотрим базовые моменты каждой из тем (*переменные, циклы, условия, математические операции и массивы*). Начнём с того, что – чтобы успешно делать задание – необходимо постоянно практиковаться в этой области. Очень сложно, не зная программирования, приступить к решению этого задания.

Базовые понятия и действия

Переменные

Переменные бывают разных типов. Сама по себе переменная это ячейка в памяти компьютера (*с каким-то именем*), которая способна хранить информацию. Самые базовые – это *целочисленные, строковые, вещественные (дробные) и логические (boolean)*. Во всех языках по-разному происходит процесс создания переменных и работа с ними. Мы будем рассматривать весь процесс на языке программирования Python.

Python обладает динамической типизацией. Что это значит? Когда приравниваем какой-либо переменной значение, то он смотрит на содержимое переменной и – в зависимости от него – присваивает определённый тип. В Python можно также применять строгую типизацию (*заранее определять тип переменной*).

```
some_var1 = 156 # Целочисленная переменная
some_var2 = 'Какой-то текст' # Строковая переменная
some_var3 = 45.4 # Вещественная (дробная) переменная
some_var3 = True # Логическая переменная (boolean)
```

Рис. 1. Варианты переменных

Всё, что находится после знака “#” является комментарием и **НЕ** воспринимается интерпретатором при запуске программы.

Следовательно, там, где записали просто число в переменную, там переменная стала целочисленной, а где записали логическое значение, там переменная стала логической.

Важная особенность логической переменной состоит в том, что она может принимать лишь два значения – «истина» и «ложь». Другими словами, **True (1)** or **False (0)**. Больше никаких значений у неё быть не может.

Математические операции

Важно иметь понимание о том, что можно делать с информацией, которая хранится в переменной, как можно взаимодействовать с переменными. В программировании существуют те же операции, что и в математике.

- + — сложение;
- — вычитание;
- * — умножение;
- / — деление.

```

some_var1 = 50
some_var2 = 40

result_1 = some_var1 + some_var2 # Результатом будет 90 (сложение)
result_2 = some_var1 - some_var2 # Результатом будет 10 (вычитание)
result_3 = some_var1 / some_var2 # Результатом будет 1.25 (деление)
result_4 = some_var1 * some_var2 # Результатом будет 2000 (умножение)

```

Рис. 2. Базовые математические операции

При записи деления переменная **result_3** будет являться вещественной (*дробной*), даже если результат деления не будет дробью. Во всех остальных математических операциях переменная будет целочисленной (*integer*). Также можно и возводить в степень и извлекать корень за счёт другой операции. Операция «**» – возведение в степень.

```

some_var1 = 36

result_1 = some_var1**2 # Возведение 36 в квадрат -> 1296
result_2 = some_var1**0.5 # Возведение 36 в степень 0.5
# (извлечение квадратного корня) -> 6.

```

Рис. 3. Возведение в степень и извлечение корня

В программировании добавляются ещё два метода деления:

mod —остаток при делении, обозначается %

div —целая часть при делении, обозначается //

```

var1 = input() # Допустим введём 40
var2 = input() # Допустим введём 50

result_1 = var1 + var2 # Результатом будет 4050
result_2 = var1 / var2 # Будет ошибка!

```

Рис. 4. Операции деления mod и div

Ввод и вывод данных

Нужно уметь и вводить (*с клавиатуры*) информацию в переменные и выводить нужную на экран. Для этого есть две определённые функции в *Python*.

input() — ввод данных;

print() — вывод данных.

Попробуем разобраться с маленькой программой на примере.

```
var1 = input() # Допустим введём 40
var2 = input() # Допустим введём 50

result_1 = var1 + var2 # Результатом будет 4050
result_2 = var1 / var2 # Будет ошибка!
```

Рис. 5. Ввод данных

Так как изначально вводимые данные при помощи функции `input()` будут являться строковыми, то при операции сложения компьютер просто соединит две строки «40» и «50», что даст в результате «4050».

Чтобы не было такой проблемы, необходимо воспользоваться операцией «приведение типов». Суть её заключается в том, что мы убеждаем компьютер, что тип переменной должен быть именно такой.

```
var1 = int(input()) # Допустим введём 40
var2 = int(input()) # Допустим введём 50

result = var1 + var2 # Результатом будет 90
```

Рис. 6. Приведение типа

Функция `int()` преобразует, в данном случае, строковое значение в числовое, за счёт этого получаем уже то сложение, которое планировали изначально.

Теперь попробуем вывести полученную сумму на экран, воспользовавшись функцией `print()`.

```
var1 = int(input()) # Допустим введём 40
var2 = int(input()) # Допустим введём 50

result = var1 + var2 # Результатом будет 90
print(result) # Выведет 90 на экран
```

Рис. 7. Вывод данных

Реализация алгоритмических конструкций

Цикл

Существуют моменты, когда нужно повторить какое-то действие определённое количество раз. Для этой задачи есть циклы, которые помогают в этом. Они есть нескольких типов, но рассмотрим только два.

Цикл **while**;

Цикл **for**.

Цикл **while** работает с предусловием. Если условие выполняется, тогда цикл работает.

```
n = 9 # Начальное значение переменной.
while n>5: # Цикл while.
    print(n) # Вывод на экран текущее значение n.
    n = n - 1 # Каждый проход цикла отнимаем от n единицу, чтобы цикл не был
              # бесконечным.
```

Рис. 8. Цикл *while*

Цикл будет работать до тех пор, пока содержимое в переменной **n** будет строго больше **5**. Результатом работы программы будет вывод четырёх чисел: **9**, **8**, **7**, **6**. Тем самым цикл сработал **четыре** раза. Повторение цикла называется **итерацией**.

Другим циклом является **for**, который выполняет действия без условия.

```
n = 10 # Начальное значение переменной.
for i in range(n): # Цикл for.
    print(i) # Вывод текущего значения переменной i.
```

Рис. 9. Цикл *for*

Особенностью данного цикла является наличие счётчика. В данном случае эта роль досталась переменной **i**. Каждую итерацию цикла счётчик увеличивается на **1** и стремится достичь значение **n** (*но не доходит до него*). Таким образом, цикл сработает **10** раз, следовательно, будет *десять итераций* цикла. Поскольку стартовое значение **i = 0**, то компьютер выведет нам десять чисел: **0, 1, 2, 3, 4, 5, 6, 7, 8, 9**.

Условие

Условие помогает делать наш алгоритм более «умным», помогает принимать ему решения на основе логических выражений. Оно помогает делать алгоритм разветвлённым. Рассмотрим одну программу и разберём её.

0	1	2	3	4
54	42	23	87	21

Рис. 10. Условие *if*

Логическое выражение $n > 20$ определяет, какое действие будет выполнено. Действие, находящиеся сразу под условием, сработает при истинном логическом выражении. А действие, которое находится после **else**, сработает при ложном значении логического выражения. Действий при ложном/истинном значении может быть несколько, не обязательно лишь одно.

Знаки логических выражений

Логические выражения могут содержать следующие знаки для условий и предусловий:

- `==` — равно: истина, если операнды равны;
- `<` — меньше: истина, если левый операнд меньше правого;
- `<=` — меньше или равно: истина, если левый операнд не больше правого;
- `>` — больше: истина, если правый операнд меньше левого;
- `>=` — больше или равно: истина, если правый операнд не больше левого;
- `!=` — не равно: истина, если операнды не равны.

Табуляция (вложенность)

В языке программирования *Python* очень важны пробелы (вложенность). Они появляются тогда, когда используем циклы и условия. Выше могли заметить отступы при работе с ними. Без них будут ошибки или программа может работать некорректно (не так, как она задумывалась создателем). При правильном написании условий, циклов, функций и так далее, отступы ставятся автоматически.

```
if n > 20: # Условие
    print('Переменная n больше 20')
```

Рис. 11. Наглядный пример отступа

Работа с массивами

Массивы являются особым типом данных. Работать с этим типом сложнее, чем с переменными. В этой статье разберём лишь маленький фрагмент работы с массивами.

Сам по себе этот тип данных представляет собой скопление однотипных элементов. У каждого элемента есть свой номер (*индекс*), по которому можно к нему обратиться и взаимодействовать с ним, как и с переменными. *Индекс* начинается с **нуля**.

0	1	2	3	4
54	42	23	87	21

Рис. 12. Пример массива

Выше приведён, состоящий из чисел, у каждого числа есть индекс (*он сверху*). Для взаимодействия с элементом массива необходимо обращаться именно к его индексу.

```
mas = [] # Создание пустого массива mas.

for i in range(5): # Цикл for
    mas.append(int(input())) # Добавление элементов (чисел) в массив.

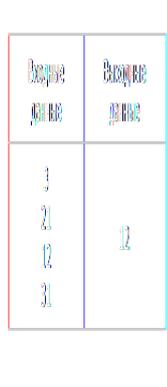
print(mas) # Вывод массива на экран.
```

Рис. 13. Работа с массивом

Здесь мы создали абсолютно пустой массив **mas**, при помощи цикла заполняем его за счёт функции **append()**. Функции **int()** и **input()** помогают вводить данные и преобразовывать строковые данные в целочисленные.

Поскольку итераций цикла у нас пять, то можем ввести **5** чисел. Допустим вводим: 5, 4, 3, 2 и 1. Тогда результатом работы программы будет массив **[1, 2, 3, 4, 5]**.

Попробуем поработать с элементами массива на похожем примере.



Индекс	Элемент
0	5
1	4
2	3
3	2
4	1

Рис. 14. Работа с элементами массива

Здесь добавляется ещё один цикл с таким же количеством повторений. Он перебирает все элементы массива и добавляет к ним число **10**.

При вводе тех же данных в результате работы программы получим следующий массив – *[11, 12, 13, 14, 15]*. Теперь можем рассмотреть само задание.

Пример задания

Напишите программу, которая в последовательности натуральных чисел определяет минимальное число, кратное 3. Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число, кратное 3. Количество чисел не превышает 1000. Введённые числа не превышают 30 000. Программа должна вывести одно число – минимальное число, кратное 3.

Пример работы программы:

Входные данные	Выходные данные
3 21 12 31	12

Рис. 15. Входные и выходные данные

Решение

Входные данные – это то, что подаётся (*вводится*) во время запуска программы. Выходные данные – это то, что нужно получить в результате работы программы при данных входных данных.

Для начала откроем интерпретатор языка программирования *Python* — **IDLE Python**.

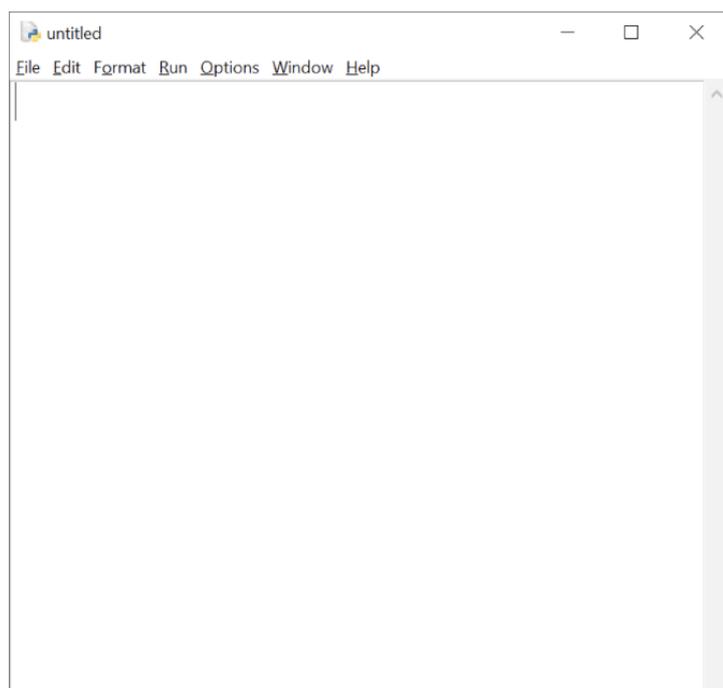


Рис. 16. Терминал языка программирования *Python*

В открытом окне (*терминал языка программирования Python*) не стоит писать программу, хоть это и возможно, но редактировать или сохранить её не получится. Поэтому откроем новый файл для написания программы.

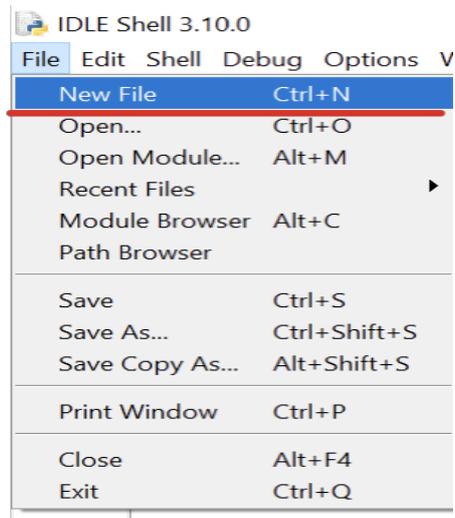


Рис. 17. Открытие нового файла

В новом файле у нас нет ничего лишнего, есть лишь пространство для написания программы.

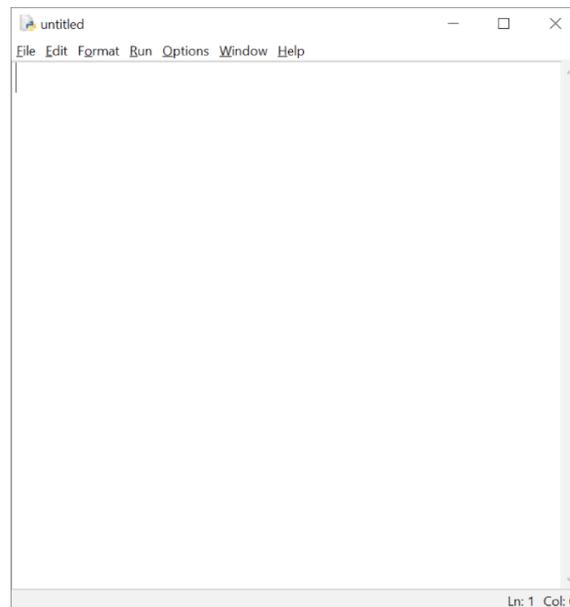


Рис. 18. Интерфейс интерпретатора Python

Теперь приступим к написанию программы, продумаем алгоритм, которым необходимо создать.

Вначале в программу поступает количество чисел для проверки. Далее вводятся числа и проверяются на кратность **трём** (*делится ли нацело*). Из всех чисел, которые кратны трём, нужно найти **минимальное** число.

Для начала напишем кусок программы для ввода данных.

```

count = int(input())
min_number = 101010
for i in range(count):
    number = int(input())

```

Рис. 19. Осуществление ввода данных

Нужно добавить переменную для поиска минимального числа и задать ей начальное значение. Когда ищем минимальное число, то такая переменная должна быть как можно больше. А когда ищем максимальное число, то как можно меньше. В условии задачи сказано, что числа не будут превышать **30000**, следовательно, могли задать начальное значение **30001**, но зададим иное.

```

count = int(input())
min_number = 101010
for i in range(count):
    number = int(input())

```

Рис. 20. Добавление «минимальной» переменной

Теперь осталось написать условие. Здесь очень важным моментом будет то, что в одном условии можем написать несколько (*сколько угодно*) логических выражений. Делается это за счёт логических связок **and (И)** и **or (ИЛИ)**.

Добавим сразу две проверки:

- Проверка на кратность;
- Проверка на минимальное число (*меньше ли новое число текущего минимального*).

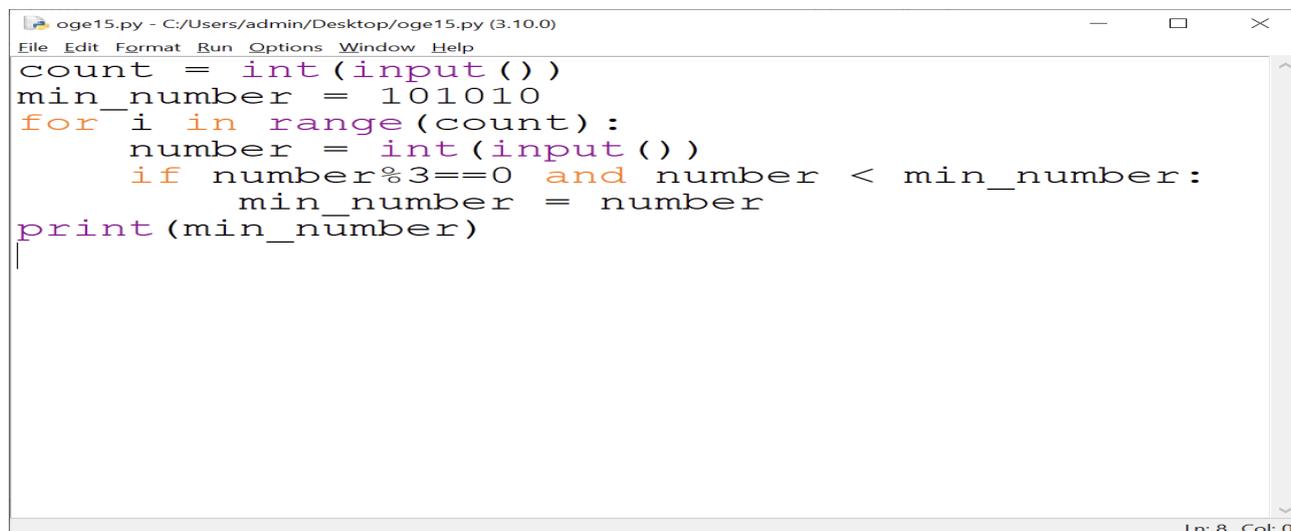
```

oge15.py - C:/Users/admin/Desktop/oge15.py (3.10.0)
File Edit Format Run Options Window Help
count = int(input())
min_number = 101010
for i in range(count):
    number = int(input())
    if number%3==0 and number < min_number:
        min_number = number
print(min_number)

```

Рис. 21. Добавление условия и вывода результата

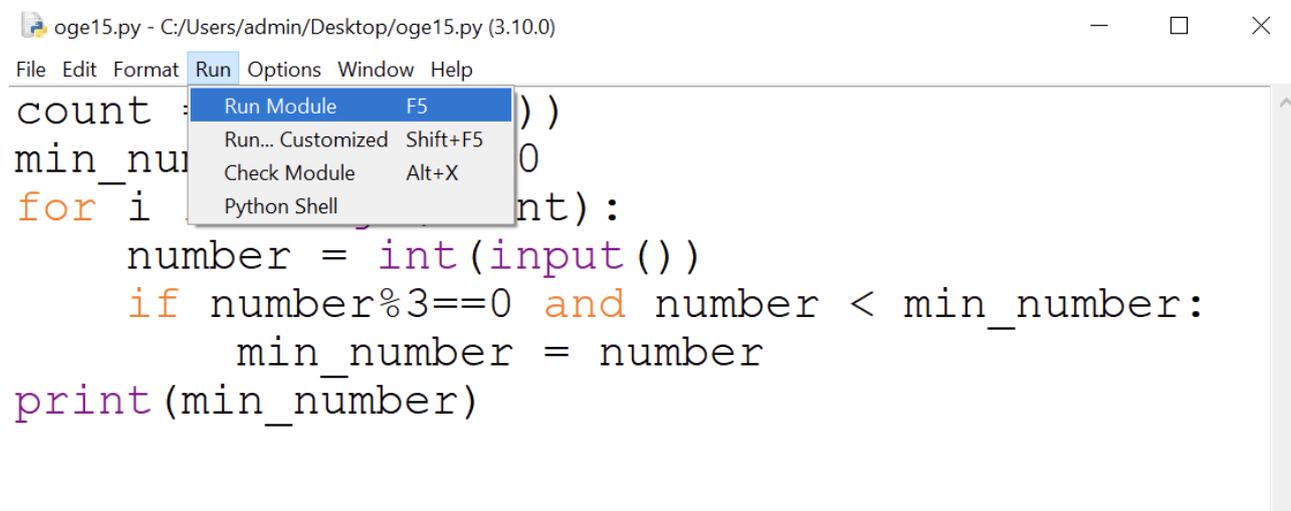
Получили законченную программу, в условии кратность проверяли за счёт остатка при делении (*операция mod*). Если остатка при делении нет, значит, число кратно.



```
oge15.py - C:/Users/admin/Desktop/oge15.py (3.10.0)
File Edit Format Run Options Window Help
count = int(input())
min_number = 101010
for i in range(count):
    number = int(input())
    if number%3==0 and number < min_number:
        min_number = number
print(min_number)
```

Рис. 22. Финальная программа

Попробуем запустить нашу программу. Для этого заходим во вкладку «Run» → «Run Module». После этого нас спросят, хотим ли сохранить, а если хотим, то куда. Без сохранения программа не сможет быть запущена.



```
oge15.py - C:/Users/admin/Desktop/oge15.py (3.10.0)
File Edit Format Run Options Window Help
count : Run Module F5
min_nu Run... Customized Shift+F5
for i Check Module Alt+X
number = int(input())
if number%3==0 and number < min_number:
    min_number = number
print(min_number)
```

Рис. 23. Запуск программы

Введём входные данные из условия задачи и посмотрим, совпадают ли выходные данные. Если будет совпадать, то скорее всего, сделали все верно.

```
>>> = RESTART: C:/Users/admin/Desktop/oge15.py
3
21
12
31
12
>>>
```

Ln: 24 Col: 0

Рис. 24. Ввод входных данных и получение выходных данных

Как видим, выходные данные совпали, минимальное число получилось **12**.

Рекомендация по проверке работы

При проверке с входными данными никогда не будет лишним попробовать ещё ряд своих данных. Эта проверка поможет точно убедиться в правильности написанной программы.

ОСОБЕННОСТИ ВЫПОЛНЕНИЯ ЗАДАНИЯ №16 ОГЭ ПО ИНФОРМАТИКЕ

Задание 16 ОГЭ по информатике проверяет умение создавать и выполнять программы для обработки числовой информации. Обычно в нём требуется:

- считать последовательность чисел (известной или неизвестной длины);
- обработать эти числа согласно условию задачи;
- вывести результат в требуемом формате.

Некоторые темы, которые необходимо изучить для решения задания:

Типы данных. Основные типы данных: числа, дробные числа, текст, перечисление через запятую или тройные апострофы.

Переменные и функции. Переменные служат для хранения информации, функции – для выполнения комплексных действий.

Операции с числами. К ним относятся сложение, вычитание, деление, умножение и возведение в степень.

Ввод данных. Функция `input()` вводит в программу данные с клавиатуры, по умолчанию принимает строку в качестве входных данных. Для ввода целых чисел используют функцию `int()`.

Вывод данных. Функция `print()` выводит данные на экран, может выводить не только значения переменных, но и значения любых выражений.

Циклы. Цикл `for` используется, когда нужно выполнить перебор элементов заранее известное количество раз. Цикл `while` применяют, когда количество вводимых чисел неизвестно, а признак окончания ввода, например, «0».

Условия. Конструкция `if – else` позволяет выполнять два блока кода: для истинного и ложного условий. Если требуется выполнение нескольких условий, используют операторы: `and` (два условия одновременно), `or` (одно из двух условий), `not` (условие не выполнялось).

Чтобы успешно выполнить задание, важно хорошо понимать основы языка программирования, особенно операторы циклов и условий, а также уметь анализировать условие задачи.

Разбор типовых заданий

Задача 1. Определение количества чисел в последовательности

Напишите программу, которая в последовательности натуральных чисел определяет количество чисел, кратных 3. Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число, кратное 3. Количество чисел не превышает 1000. Введённые числа не превышают 30 000. Программа должна вывести одно число - количество чисел, кратных 3.

Пример работы программы:

Входные данные	Выходные данные
4 15 26 24 7	2

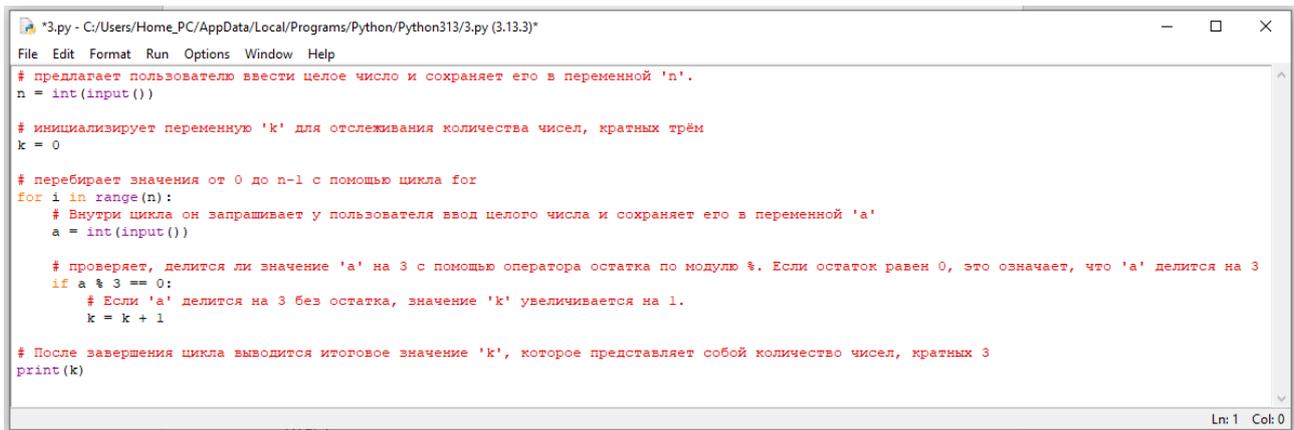
Решение:

```
n=int(input())
k=0
for i in range(n):
    a=int(input())
    if a%3==0:
        k=k+1

print(k)
```

Решение с пояснением:

```
# предлагает пользователю ввести целое число и сохраняет его в
переменной 'n'.
n=int(input())
# инициализирует переменную 'k' для отслеживания количества чисел,
кратных трём
k=0
# перебирает значения от 0 до n-1 с помощью цикла for
for i in range(n):
    # Внутри цикла он запрашивает у пользователя ввод целого числа и
сохраняет его в переменной 'a'
    a=int(input())
    # проверяет, делится ли значение 'a' на 3 с помощью оператора остатка по
модулю %. Если остаток равен 0, это означает, что 'a' делится на 3
    if a%3==0:
        # Если 'a' делится на 3 без остатка, значение 'k' увеличивается на 1.
        k=k+1
    # После завершения цикла выводится итоговое значение 'k', которое
представляет собой количество чисел, кратных 3
print(k)
```



```
*3.py - C:/Users/Home_PC/AppData/Local/Programs/Python/Python313/3.py (3.13.3)*
File Edit Format Run Options Window Help
# предлагает пользователю ввести целое число и сохраняет его в переменной 'n'.
n = int(input())

# инициализирует переменную 'k' для отслеживания количества чисел, кратных трём
k = 0

# перебирает значения от 0 до n-1 с помощью цикла for
for i in range(n):
    # Внутри цикла он запрашивает у пользователя ввод целого числа и сохраняет его в переменной 'a'
    a = int(input())

    # проверяет, делится ли значение 'a' на 3 с помощью оператора остатка по модулю %. Если остаток равен 0, это означает, что 'a' делится на 3
    if a % 3 == 0:
        # Если 'a' делится на 3 без остатка, значение 'k' увеличивается на 1.
        k = k + 1

# После завершения цикла выводится итоговое значение 'k', которое представляет собой количество чисел, кратных 3
print(k)
```

Рис. 25. Пример программы

Краткое пояснение выполнения программы

В первой строчке мы запрашиваем число n . Переменная k отвечает за количество найденных чисел, удовлетворяющих условию задачи. В начале количество найденных чисел равно нулю.

Затем идёт ЦИКЛ **for**. Цикл повторяется n раз. Внутри цикла запрашиваем число a . Это очередное число в нашей последовательности. Это число проверяется с помощью условия **if** (переводится, как **ЕСЛИ**). Если остаток от деления на 3 равен нулю, то число a делится на 3. Операция $\%$ – это и есть остаток от деления.

Если условие будет истинным, то будет выполняться строчка $k=k+1$. Т.е., если мы нашли число, которое кратно 3, то прибавляем единицу к счётчику k . Если условие не выполняется (ложно), то строка $k=k+1$ не выполняется.

После того, как весь цикл прокрутится, в переменной k будет сидеть ответ.

Выполните код и проверьте тестовую таблицу, которая дана в условии задачи.

Задача 2. Определение количества чисел в последовательности, оканчивающихся на N

Напишите программу, которая в последовательности натуральных чисел определяет количество чисел, оканчивающихся на 4. Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число, оканчивающееся на 4. Количество чисел не превышает 1000. Введённые числа не превышают 30 000. Программа должна вывести одно число - количество чисел, оканчивающихся на 4.

Пример работы программы:

Входные данные	Выходные данные
3 26 33 24	1

Решение:

```
n=int(input())
k=0
for i in range(n):
    a=int(input())
    if a%10==4:
        k=k+1

print(k)
```

Решение с пояснением:

```
# Первая строка n=int(input()) запрашивает у пользователя ввод целого числа
# 'n', которое обозначает количество целых чисел, которые будут введены
n=int(input())
# Во второй строке k=0 инициализируется переменная-счётчик k для отсле-
# живания количества целых чисел, которые оканчиваются на цифру 4
k=0
# for Цикл for i in range(n): повторяется 'n' раз, где 'n' – количество целых
# чисел, которые пользователь введет.
for i in range(n):
    # Внутри цикла строка a=int(input()) запрашивает у пользователя ввод следу-
    # ющего целого числа
    a=int(input())
    # if Оператор if a%10==4: проверяет, равна ли последняя цифра целого числа
    # 'a' 4. Оператор % используется для получения остатка от деления 'a' на 10, что
    # дает последнюю цифру числа
    if a%10==4:
        # Если последняя цифра равна 4, строка k=k+1 увеличивает переменную
        # счётчика 'k' на 1
        k=k+1
# После завершения цикла с помощью оператора k выводится итоговое зна-
# чение print(k)
print(k)
```

```

4.py - C:/Users/Home_PC/AppData/Local/Programs/Python/Python313/4.py (3.13.3)
File Edit Format Run Options Window Help
# Первая строка n=int(input()) запрашивает у пользователя ввод целого числа 'n', которое обозначает количество целых чисел, которые будут введены
n=int(input())
# Во второй строке k=0 инициализируется переменная-счётчик k для отслеживания количества целых чисел, которые оканчиваются на цифру 4
k=0
# Цикл for i in range(n): повторяется 'n' раз, где 'n' - количество целых чисел, которые пользователь введет.
for i in range(n):
# Внутри цикла строка a=int(input()) запрашивает у пользователя ввод следующего целого числа
a=int(input())
# if Оператор if a%10==4: проверяет, равна ли последняя цифра целого числа 'a' 4. Оператор % используется для получения остатка от деления 'a' на 10, что даёт последнюю цифру числа
if a%10==4:
# Если последняя цифра равна 4, строка k=k+1 увеличивает переменную счётчика 'k' на 1
k=k+1
# После завершения цикла с помощью оператора k выводится итоговое значение print(k)
print(k)
Ln: 15 Col: 0

```

Рис. 26. Пример программы

Эта программа отличается от предыдущей тем, что нужно найти количество чисел, которые оканчиваются на цифру 4.

Для того, чтобы найти, на что **оканчивается положительное число**, достаточно взять **остаток от деления на 10**.

Пример:

$$16 : 10 = 1 \text{ (остаток } 6) \quad 28 : 10 = 2 \text{ (остаток } 8) \quad 432 : 10 = 43 \text{ (остаток } 2)$$

Задача 3. Определение суммы элементов последовательности с учетом сложного условия

Напишите программу, которая в последовательности натуральных чисел определяет сумму чисел, кратных 3 и оканчивающихся на 1. Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число, кратное 3 и оканчивающееся на 1. Количество чисел не превышает 100. Введённые числа не превышают 300. Программа должна вывести одно число - сумму чисел, кратных 3 и оканчивающихся на 1.

Пример работы программы:

Входные данные	Выходные данные
5	6603
21	
51	
6531	
121	
22	

Решение:

```

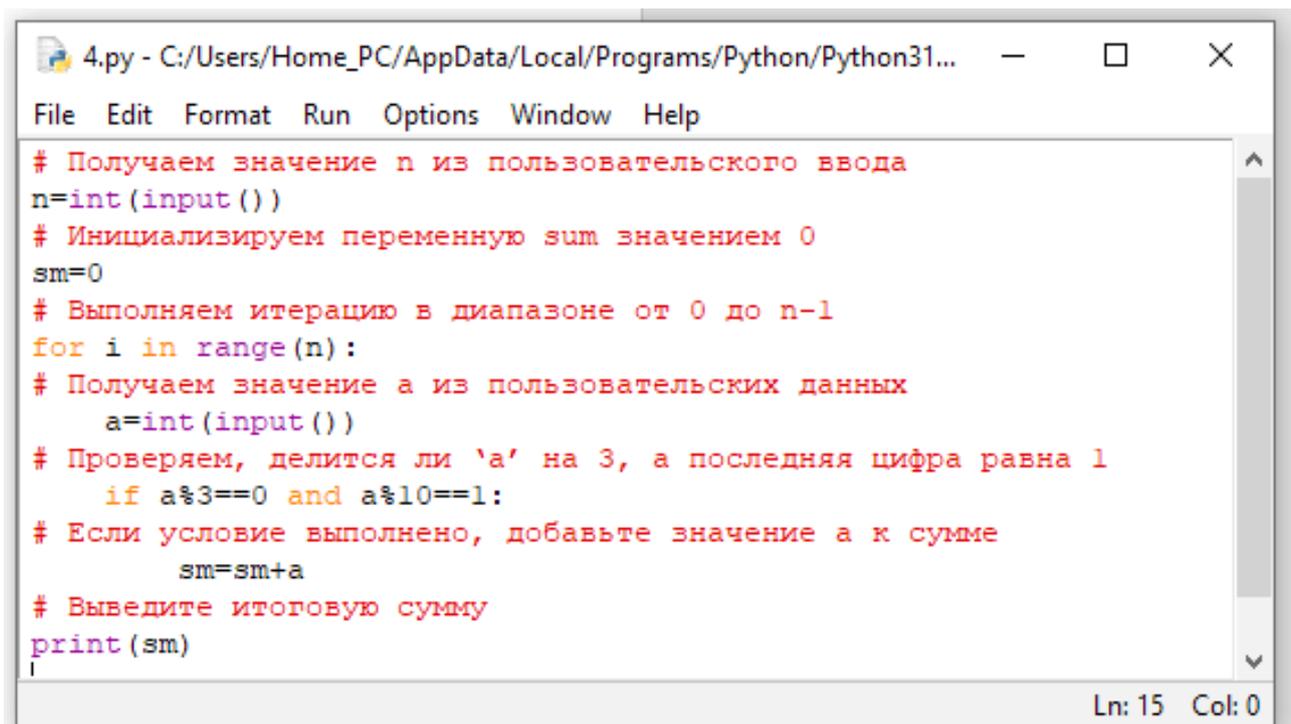
n=int(input())
sm=0
for i in range(n):
a=int(input())
if a%3==0 and a%10==1:
sm=sm+a

print(sm)

```

Решение с пояснением:

```
# Получаем значение n из пользовательского ввода
n=int(input())
# Инициализируем переменную sum значением 0
sm=0
# Выполняем итерацию в диапазоне от 0 до n-1
for i in range(n):
# Получаем значение a из пользовательских данных
    a=int(input())
# Проверяем, делится ли 'a' на 3, а последняя цифра равна 1
    if a%3==0 and a%10==1:
# Если условие выполнено, добавьте значение a к сумме
        sm=sm+a
# Выведите итоговую сумму
print(sm)
```

A screenshot of a Python IDE window titled "4.py - C:/Users/Home_PC/AppData/Local/Programs/Python/Python31...". The window contains the same Python code as shown in the previous block. The code is color-coded: comments are red, keywords like 'int', 'input', 'range', 'if', 'and', 'print' are purple, and variables like 'n', 'sm', 'a', 'i' are blue. The status bar at the bottom right shows "Ln: 15 Col: 0".

```
4.py - C:/Users/Home_PC/AppData/Local/Programs/Python/Python31...
File Edit Format Run Options Window Help
# Получаем значение n из пользовательского ввода
n=int(input())
# Инициализируем переменную sum значением 0
sm=0
# Выполняем итерацию в диапазоне от 0 до n-1
for i in range(n):
# Получаем значение a из пользовательских данных
    a=int(input())
# Проверяем, делится ли 'a' на 3, а последняя цифра равна 1
    if a%3==0 and a%10==1:
# Если условие выполнено, добавьте значение a к сумме
        sm=sm+a
# Выведите итоговую сумму
print(sm)
|
Ln: 15 Col: 0
```

Рис. 27. Пример программы

За сумму отвечает переменная **sm**.

Здесь нужно, чтобы два условия выполнялись для числа **a**. Это достигается за счёт союза **and** (союз И).

Когда ищем сумму, мы уже к переменной **sm** прибавляем не 1, а само число **a**.

Задача 4. Определение суммы чисел в заданном интервале с учетом сложного условия

Напишите программу, которая в последовательности натуральных чисел определяет сумму трехзначных нечётных чисел, кратных 7. Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется трёхзначное нечётное число, кратное 7. Количество чисел не превышает 100. Введённые числа не превышают 300. Программа должна вывести одно число - сумму трёхзначных нечётных чисел, кратных 7.

Пример работы программы:

Входные данные	Выходные данные
5	896
441	
455	
294	
121	
2261	

Решение:

```
n=int(input())
sm=0
for i in range(n):
    a=int(input())
    if a%2!=0 and a%7==0 and a > 99 and a < 1000:
        sm=sm+a

print(sm)
```

Решение с пояснением:

```
# Код начинается с того, что пользователю предлагается ввести целочисленное значение, которое хранится в переменной 'n'
n=int(input())
# Переменная инициализируется значением 0, которое будет использоваться для хранения суммы допустимых чисел 'sm'
sm=0
# Затем код входит в цикл с перебором раз, где в каждой итерации пользователю предлагается ввести еще одно целочисленное значение, которое хранится в переменной 'n' 'a'
for i in range(n):
    a=int(input())
# Внутри цикла код проверяет, удовлетворяет ли число следующим условиям: 'a'
```

```

# является нечетным (т.е., a % 2 != 0)
# делится на 7 (т.е. a % 7 == 0)
# составляет от 99 до 1000 (включительно) (т.е. 99 < a < 1000)
    if a%2!=0 and a%7==0 and a > 99 and a < 1000:
# Если число удовлетворяет всем условиям, оно прибавляется к переменной
'a' 'sm'
        sm=sm+a
# После завершения цикла выводится окончательное значение of 'sm'
print(sm)

```

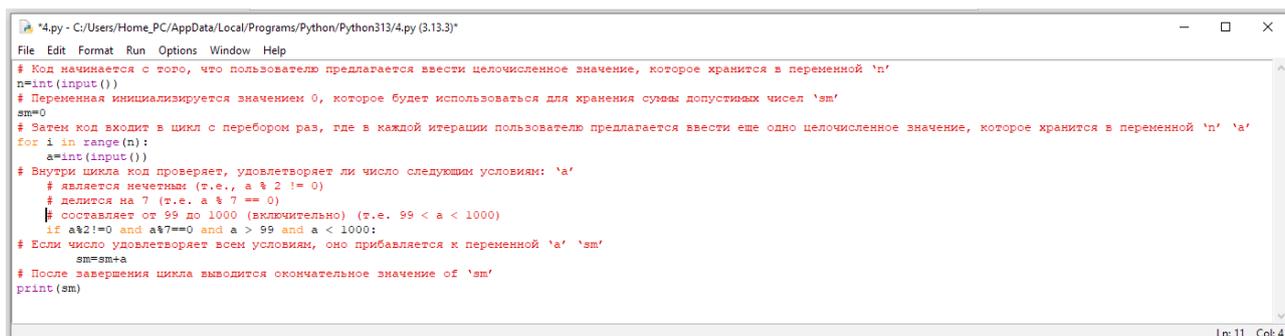


Рис. 28. Пример программы

Нечётное число даёт при делении на 2 остаток, не равный нулю. Знак «неравно» пишется так: !=. Трёхзначные числа легко отследить, т.к. эти числа больше 99, но меньше 1000.

Эти все требования оформляем в виде сложного условия через союз И.

Задача 5. Определение максимального числа в последовательности с учетом заданного условия

Напишите программу, которая в последовательности натуральных чисел определяет максимальное число, кратное 8. Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число, кратное 8. Количество чисел не превышает 1000. Введённые числа не превышают 30 000. Программа должна вывести одно число - максимальное число, кратное 8.

Пример работы программы:

Входные данные	Выходные данные
4	64
56	
431	
64	
90	

Решение:

```
n=int(input())
mx=0
for i in range(n):
    a=int(input())
    if a%8==0:
        mx=max(mx, a)
print(mx)
```

Решение с пояснением:

Код начинается с получения от пользователя количества целых чисел n с помощью функции input() и преобразования его в целое число с помощью int()

```
n=int(input())
```

Переменная 'mx' инициализируется значением 0, которое будет использоваться для хранения максимального значения, делимого на 8

```
mx=0
```

Затем код переходит в цикл, который выполняется 'n' раз, где i принимает значения от 0 до n-1

```
for i in range(n):
```

Внутри цикла следующее целое число 'a' получается от пользователя с помощью функции input() и преобразуется в целое число с помощью int()

```
    a=int(input())
```

Затем код проверяет, делится ли целое число a на 8 с помощью оператора остатка от деления %. Если остаток равен 0, это означает, что 'a' делится на 8

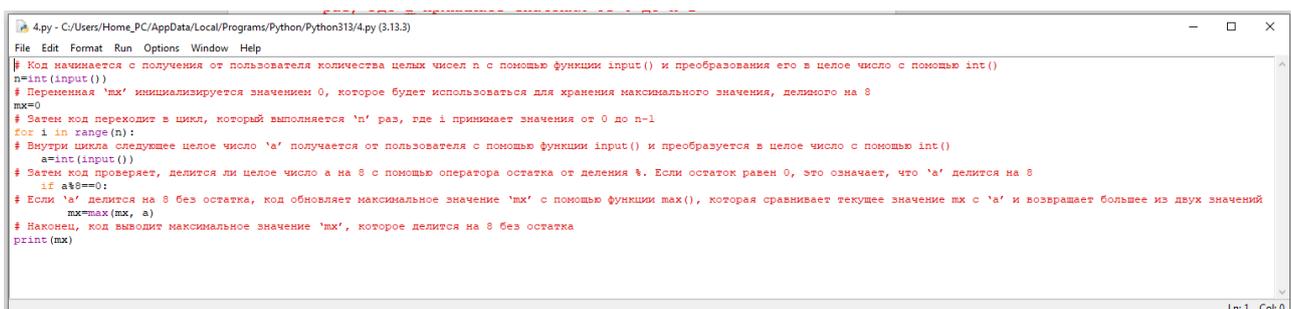
```
    if a%8==0:
```

Если 'a' делится на 8 без остатка, код обновляет максимальное значение 'mx' с помощью функции max(), которая сравнивает текущее значение mx с 'a' и возвращает большее из двух значений

```
        mx=max(mx, a)
```

Наконец, код выводит максимальное значение 'mx', которое делится на 8 без остатка

```
print(mx)
```



The image shows a screenshot of a Python IDE window titled '4.py - C:/Users/Home_PC/AppData/Local/Programs/Python/Python313/4.py (3.13.3)'. The window contains the following code:

```
# Код начинается с получения от пользователя количества целых чисел n с помощью функции input() и преобразования его в целое число с помощью int()
n=int(input())
# Переменная 'mx' инициализируется значением 0, которое будет использоваться для хранения максимального значения, делимого на 8
mx=0
# Затем код переходит в цикл, который выполняется 'n' раз, где i принимает значения от 0 до n-1
for i in range(n):
# Внутри цикла следующее целое число 'a' получается от пользователя с помощью функции input() и преобразуется в целое число с помощью int()
    a=int(input())
# Затем код проверяет, делится ли целое число a на 8 с помощью оператора остатка от деления %. Если остаток равен 0, это означает, что 'a' делится на 8
    if a%8==0:
# Если 'a' делится на 8 без остатка, код обновляет максимальное значение 'mx' с помощью функции max(), которая сравнивает текущее значение mx с 'a' и возвращает большее из двух значений
        mx=max(mx, a)
# Наконец, код выводит максимальное значение 'mx', которое делится на 8 без остатка
print(mx)
```

Рис. 29. Пример программы

Здесь уже заводим переменную **mx** для максимального значения, которое делится на 8. Она и пойдёт в ответ.

Нас интересуют только числа, которые делятся на 8. Если это так, числа **a** претендуют, чтобы попасть в переменную **mx**. Функция **max()** выбирает среди двух значений наибольшее. Если **a** больше, чем значение, которое находится в **mx** в данный момент времени, то оно перезапишется в переменную **mx**.

В начале в **mx** самое маленькое число **0**, чтобы его легко было «победить». Последующее число будет «победить» уже сложнее. После прохождения всего цикла в переменной **mx** будет находиться максимальное число.

Задача 6. Определение минимального числа в последовательности с учетом заданного условия

Напишите программу, которая в последовательности натуральных чисел определяет минимальное число, оканчивающееся на 6. Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число, оканчивающееся на 6. Количество чисел не превышает 1000. Введённые числа не превышают 30 000. Программа должна вывести одно число – минимальное число, оканчивающееся на 6.

Пример работы программы:

Входные данные	Выходные данные
4	56
56	
500	
86	
13	

Решение:

```
n=int(input())
mn=30001
for i in range(n):
    a=int(input())
    if a%10==6:
        mn=min(mn, a)
print(mn)
```

Решение с пояснением:

запрашивает у пользователя ввод целого числа и сохраняет его в переменной 'n'

```

n=int(input())
# присваивает переменной 'mn' большое значение 30001
mn=30001
# Затем он выполняет итерацию по диапазону 'n' значений
for i in range(n):
# Он проверяет, равна ли последняя цифра 'a'
    a=int(input())
# Если да, то он обновляет значение 'mn' до минимума из текущего 'mn' и 'a'
    if a%10==6:
        mn=min(mn, a)
# Наконец, он выводит окончательное значение 'mn'
print(mn)
# Цель этого кода – найти наименьшее целое число среди значений n,
введённых пользователем, где последняя цифра целого числа равна 6.

```

Рис. 30. Пример программы

Эта программа похожа на поиск максимального числа. Существенная разница заключается в том, что при поиске минимального значения в переменную **mn** кладём, наоборот, самое большое значение. Если мы оставим ноль, то его никто не сможет «победить», и в ответе получится тоже ноль.

Задача 7. Учет признака окончания последовательности

Напишите программу, которая в последовательности натуральных чисел вычисляет сумму всех двузначных чисел, кратных 5. Программа получает на вход натуральные числа, количество введённых чисел неизвестно, последовательность чисел заканчивается числом 0 (0 – признак окончания ввода, не входит в последовательность). Количество чисел не превышает 1000. Введённые числа не превышают 30 000. Программа должна вывести одно число: сумму всех двузначных чисел, кратных 5.

Пример работы программы:

Входные данные	Выходные данные
55 120 25 68 0	80

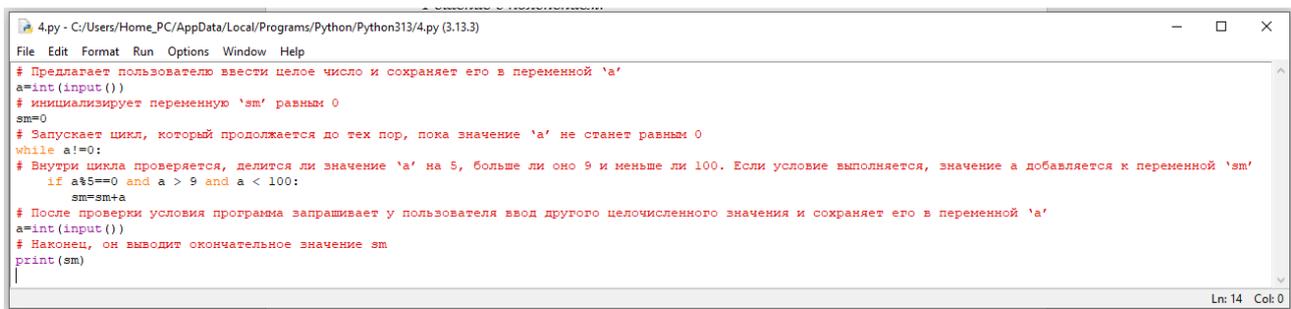
Решение:

```
a=int(input())
sm=0
while a!=0:
    if a%5==0 and a > 9 and a < 100:
        sm=sm+a
    a=int(input())

print(sm)
```

Решение с пояснением:

```
# Предлагает пользователю ввести целое число и сохраняет его в переменной
'a'
a=int(input())
# инициализирует переменную 'sm' равным 0
sm=0
# Запускает цикл, который продолжается до тех пор, пока значение 'a' не
станет равным 0
while a!=0:
    # Внутри цикла проверяется, делится ли значение 'a' на 5, больше ли оно 9 и
меньше ли 100. Если условие выполняется, значение a добавляется к переменной
'sm'
    if a%5==0 and a > 9 and a < 100:
        sm=sm+a
    # После проверки условия программа запрашивает у пользователя ввод
другого целочисленного значения и сохраняет его в переменной 'a'
    a=int(input())
# Наконец, он выводит окончательное значение sm
print(sm)
```



```
4.py - C:/Users/Home_PC/AppData/Local/Programs/Python/Python313/4.py (3.13.3)
File Edit Format Run Options Window Help
# Предлагает пользователю ввести целое число и сохраняет его в переменной 'a'
a=int(input())
# инициализирует переменную 'sm' равным 0
sm=0
# Запускает цикл, который продолжается до тех пор, пока значение 'a' не станет равным 0
while a!=0:
# Внутри цикла проверяется, делится ли значение 'a' на 5, больше ли оно 9 и меньше ли 100. Если условие выполняется, значение a добавляется к переменной 'sm'
    if a%5==0 and a > 9 and a < 100:
        sm=sm+a
# После проверки условия программа запрашивает у пользователя ввод другого целочисленного значения и сохраняет его в переменной 'a'
a=int(input())
# Наконец, он выводит окончательное значение sm
print(sm)
```

Рис. 31. Пример программы

Здесь уже нет переменной **n** (количества чисел в последовательности). Как пользователю надоеет вводить числа, он должен ввести число ноль.

Вместо **n** запрашиваем число **a**. Заводим переменную **sm** для ответа. Потом начинается ЦИКЛ **while**. Он будет выполняться (повторяться), пока условие, стоящее справа, истинное. Мы там прописываем, пока число **a** не равно нулю, мы продолжаем, иначе это окончание последовательности.

Внутри ЦИКЛА проверяем число **a** согласно условию задачи.

После проверки, запрашиваем новое число **a** (новый элемент последовательности).

СБОРНИК ЗАДАЧ С РЕШЕНИЯМИ ИЗ ОТКРЫТОГО БАНКА ТЕСТОВЫХ ЗАДАНИЙ С САЙТА ФГБНУ «ФЕДЕРАЛЬНЫЙ ИНСТИТУТ ПЕДАГОГИЧЕСКИХ ИЗМЕРЕНИЙ»

Примечание. Файлы программ представлены в электронном приложении

Номер: F1B54B

Напишите программу, которая в последовательности натуральных десятичных чисел определяет количество элементов, запись которых в системе счисления с основанием 7 оканчивается цифрой 1. Если среди входных данных таких элементов нет, программа должна вывести «NO».

Программа получает на вход в первой строке натуральное число – количество чисел N ($3 \leq N \leq 10\,000$), затем N натуральных чисел, не превышающих 30 000, каждое в отдельной строке.

Программа должна вывести одно число – количество десятичных чисел (элементов последовательности), запись которых в 7-ричной системе счисления оканчивается цифрой 1, или «NO», если среди входных данных таких элементов нет.

Пример работы программы

Входные данные	Выходные данные
6	3
12	
15	
10	
71	
50	
11	

Номер: 0A474B

Напишите программу, которая в последовательности натуральных десятичных чисел определяет наибольший элемент, запись которого в системе счисления с основанием 7 оканчивается **нечётной** цифрой. Если среди входных данных таких элементов нет, программа должна вывести «NO».

Программа получает на вход в первой строке натуральное число – количество чисел N ($3 \leq N \leq 10\,000$), затем N натуральных чисел, не превышающих 30 000, каждое в отдельной строке.

Программа должна вывести одно десятичное число – наибольший элемент последовательности, запись которого в 7-ричной системе счисления оканчивается **нечётной** цифрой, или «NO», если среди входных данных таких элементов нет.

Пример работы программы

Входные данные	Выходные данные
6 12 15 10 71 50 11	71

Номер: 0FCA49

Напишите программу, которая в последовательности натуральных десятичных чисел определяет среднее арифметическое элементов, запись которых в системе счисления с основанием 7 оканчивается **нечётной** цифрой. Если среди входных данных таких элементов нет, программа должна вывести «NO».

Программа получает на вход в первой строке натуральное число – количество чисел N ($3 \leq N \leq 10\,000$), затем N натуральных чисел, не превышающих 30 000, каждое в отдельной строке.

Программа должна вывести одно число – среднее арифметическое десятичных чисел (элементов последовательности), запись которых в 7-ричной системе счисления оканчивается **нечётной** цифрой, или «NO», если среди входных данных таких элементов нет.

Пример работы программы

Входные данные	Выходные данные
6 12 15 10 71 50 11	31.6

Номер: 11484C

Напишите программу, которая в последовательности натуральных десятичных чисел определяет количество элементов, запись которых в системе счисления с основанием 7 оканчивается цифрой 2. Если среди входных данных таких элементов нет, программа должна вывести «NO».

Программа получает на вход в первой строке натуральное число – количество чисел N ($3 \leq N \leq 10\,000$), затем N натуральных чисел, не превышающих 30 000, каждое в отдельной строке.

Программа должна вывести одно число – количество десятичных чисел (элементов последовательности), запись которых в 7-ричной системе счисления оканчивается цифрой 2, или «NO», если среди входных данных таких элементов нет.

Пример работы программы

Входные данные	Выходные данные
5	2
37	
22	
16	
25	
42	

Номер: CC7540

Напишите программу, которая в последовательности натуральных десятичных чисел определяет наибольший элемент, запись которого в системе счисления с основанием 7 оканчивается **нечётной** цифрой. Если среди входных данных таких элементов нет, программа должна вывести «NO».

Программа получает на вход натуральные десятичные числа, не превышающие 30 000, каждое в отдельной строке. Количество введённых чисел неизвестно, их последовательность заканчивается числом 0 (0 – признак окончания ввода, в последовательность не входит).

Программа должна вывести одно десятичное число – наибольший элемент последовательности, запись которого в 7-ричной системе счисления оканчивается **нечётной** цифрой, или «NO», если среди входных данных таких элементов нет.

Пример работы программы

Входные данные	Выходные данные
12	71
15	
10	
71	
50	
11	
0	

Номер: AB87F8

Напишите программу, которая в последовательности натуральных десятичных чисел определяет количество элементов, запись которых в системе счисления с основанием 5 оканчивается **нечётной** цифрой. Если среди входных данных таких элементов нет, программа должна вывести «NO».

Программа получает на вход в первой строке натуральное число – количество чисел N ($3 \leq N \leq 10\,000$), затем N натуральных чисел, не превышающих 30 000, каждое в отдельной строке.

Программа должна вывести одно число – количество десятичных чисел (элементов последовательности), запись которых в 5-ричной системе счисления оканчивается **нечётной** цифрой, или «NO», если среди входных данных таких элементов нет.

Пример работы программы

Входные данные	Выходные данные
6 12 15 10 71 50 11	2

Номер: 014503

Напишите программу, которая в последовательности натуральных десятичных чисел определяет среднее арифметическое элементов, запись которых в системе счисления с основанием 5 оканчивается **нечётной** цифрой. Если среди входных данных таких элементов нет, программа должна вывести «NO».

Программа получает на вход натуральные десятичные числа, не превышающие 30 000, каждое в отдельной строке. Количество введённых чисел неизвестно, их последовательность заканчивается числом 0 (0 – признак окончания ввода, в последовательность не входит).

Программа должна вывести одно число – среднее арифметическое десятичных чисел (элементов последовательности), запись которых в системе счисления с основанием 5 оканчивается **нечётной цифрой**, или «NO», если среди входных данных таких элементов нет.

Пример работы программы

Входные данные	Выходные данные
12 15 10 71 50 11 0	41

Номер: 026108

Напишите программу, которая в последовательности натуральных десятичных чисел определяет среднее арифметическое элементов, запись которых в системе счисления с основанием 7 оканчивается цифрой 5. Если среди входных данных таких элементов нет, программа должна вывести «NO».

Программа получает на вход натуральные десятичные числа, не превышающие 30 000, каждое в отдельной строке. Количество введенных чисел неизвестно, их последовательность заканчивается числом 0 (0 – признак окончания ввода, в последовательность не входит).

Программа должна вывести одно число – среднее арифметическое десятичных чисел (элементов последовательности), запись которых в 7-ричной системе счисления оканчивается цифрой 5, или «NO», если среди входных данных таких элементов нет.

Пример работы программы

Входные данные	Выходные данные
12	40
14	
20	
68	
25	
0	

Номер: 41D07B

Напишите программу, которая в последовательности натуральных десятичных чисел определяет среднее арифметическое элементов, запись которых в системе счисления с основанием 7 оканчивается цифрой 5. Если среди входных данных таких элементов нет, программа должна вывести «NO».

Программа получает на вход в первой строке натуральное число – количество чисел N ($3 \leq N \leq 10\,000$), затем N натуральных чисел, не превышающих 30 000, каждое в отдельной строке.

Программа должна вывести одно число – среднее арифметическое десятичных чисел (элементов последовательности), запись которых в 7-ричной системе счисления оканчивается цифрой 5, или «NO», если среди входных данных таких элементов нет.

Пример работы программы

Входные данные	Выходные данные
5	40
12	
14	
20	
68	
25	

Номер: 50D7B2

Напишите программу, которая в последовательности натуральных десятичных чисел определяет среднее арифметическое элементов, запись которых в системе счисления с основанием 7 оканчивается цифрой 4. Если среди входных данных таких элементов нет, программа должна вывести «NO».

Программа получает на вход в первой строке натуральное число – количество чисел N ($3 \leq N \leq 10\,000$), затем N натуральных чисел, не превышающих 30 000, каждое в отдельной строке.

Программа должна вывести одно число – среднее арифметическое десятичных чисел (элементов последовательности), запись которых в 7-ричной системе счисления оканчивается цифрой 4, или «NO», если среди входных данных таких элементов нет.

Пример работы программы

Входные данные	Выходные данные
5 11 25 30 14 54	18

Номер: 0B7312

Напишите программу, которая в последовательности натуральных десятичных чисел определяет сумму элементов, запись которых в системе счисления с основанием 7 оканчивается цифрой 3. Если среди входных данных таких элементов нет, программа должна вывести «NO».

Программа получает на вход натуральные десятичные числа, не превышающие 30 000, каждое в отдельной строке. Количество введённых чисел неизвестно, их последовательность заканчивается числом 0 (0 – признак окончания ввода, в последовательность не входит).

Программа должна вывести одно число – сумму десятичных чисел (элементов последовательности), запись которых в 7-ричной системе счисления оканчивается цифрой 3, или «NO», если среди входных данных таких элементов нет.

Пример работы программы

Входные данные	Выходные данные
33 63 17 24 53 0	41

Номер: 4CEE2A

Напишите программу, которая в последовательности натуральных десятичных чисел определяет сумму элементов, запись которых в системе счисления с основанием 5 оканчивается чётной цифрой. Если среди входных данных таких элементов нет, программа должна вывести «NO».

Программа получает на вход натуральные десятичные числа, не превышающие 30 000, каждое в отдельной строке. Количество введённых чисел неизвестно, их последовательность заканчивается числом 0 (0 – признак окончания ввода, в последовательность не входит).

Программа должна вывести одно число – сумму десятичных чисел (элементов последовательности), запись которых в системе счисления с основанием 5 оканчивается чётной цифрой, или «NO», если среди входных данных таких элементов нет.

Пример работы программы

Входные данные	Выходные данные
12	87
15	
10	
71	
50	
11	
0	

Номер: 9A0EDE

Напишите программу, которая в последовательности натуральных десятичных чисел определяет количество элементов, запись которых в системе счисления с основанием 5 оканчивается нечётной цифрой. Если среди входных данных таких элементов нет, программа должна вывести «NO».

Программа получает на вход натуральные десятичные числа, не превышающие 30 000, каждое в отдельной строке. Количество введённых чисел неизвестно, их последовательность заканчивается числом 0 (0 – признак окончания ввода, в последовательность не входит).

Программа должна вывести одно число – количество десятичных чисел (элементов последовательности), запись которых в системе счисления с основанием 5 оканчивается нечётной цифрой, или «NO», если среди входных данных таких элементов нет.

Пример работы программы

Входные данные	Выходные данные
12	2
15	
10	
71	
50	
11	
0	

Номер: BDAD52

Напишите программу, которая в последовательности натуральных десятичных чисел определяет среднее арифметическое элементов, запись которых в системе счисления с основанием 7 оканчивается **нечётной** цифрой. Если среди входных данных таких элементов нет, программа должна вывести «NO».

Программа получает на вход натуральные десятичные числа, не превышающие 30 000, каждое в отдельной строке. Количество введённых чисел неизвестно, их последовательность заканчивается числом 0 (0 – признак окончания ввода, в последовательность не входит).

Программа должна вывести одно число – среднее арифметическое десятичных чисел (элементов последовательности), запись которых в системе счисления с основанием 7 оканчивается **нечётной цифрой**, или «NO», если среди входных данных таких элементов нет.

Пример работы программы

Входные данные	Выходные данные
12	31.6
15	
10	
71	
50	
11	
0	

Номер: E124A8

Напишите программу, которая в последовательности натуральных десятичных чисел определяет сумму элементов, запись которых в системе счисления с основанием 5 оканчивается чётной цифрой. Если среди входных данных таких элементов нет, программа должна вывести «NO».

Программа получает на вход в первой строке натуральное число – количество чисел N ($3 \leq N \leq 10\,000$), затем N натуральных чисел, не превышающих 30 000, каждое в отдельной строке.

Программа должна вывести одно число – сумму десятичных чисел (элементов последовательности), запись которых в 5-ричной системе счисления оканчивается чётной цифрой, или «NO», если среди входных данных таких элементов нет.

Пример работы программы

Входные данные	Выходные данные
6	87
12	
15	
10	
71	
50	
11	

Номер: DBFBCC

Напишите программу, которая в последовательности натуральных десятичных чисел определяет количество элементов, запись которых в системе счисления с основанием 7 оканчивается цифрой 1. Если среди входных данных таких элементов нет, программа должна вывести «NO».

Программа получает на вход натуральные десятичные числа, не превышающие 30 000, каждое в отдельной строке. Количество введённых чисел неизвестно, их последовательность заканчивается числом 0 (0 – признак окончания ввода, в последовательность не входит).

Программа должна вывести одно число – количество десятичных чисел (элементов последовательности), запись которых в 7-ричной системе счисления оканчивается цифрой 1, или «NO», если среди входных данных таких элементов нет.

Пример работы программы

Входные данные	Выходные данные
12	3
15	
10	
71	
50	
11	
0	

Номер: E4F7CA

Напишите программу, которая в последовательности натуральных десятичных чисел определяет сумму элементов, запись которых в системе счисления с основанием 7 оканчивается цифрой 6. Если среди входных данных таких элементов нет, программа должна вывести «NO».

Программа получает на вход натуральные десятичные числа, не превышающие 30 000, каждое в отдельной строке. Количество введённых чисел неизвестно, их последовательность заканчивается числом 0 (0 – признак окончания ввода, в последовательность не входит).

Программа должна вывести одно число – сумму десятичных чисел (элементов последовательности), запись которых в 7-ричной системе счисления оканчивается цифрой 6, или «NO», если среди входных данных таких элементов нет.

Пример работы программы

Входные данные	Выходные данные
20	33
13	
9	
28	
12	
0	

Номер: E4F7CA

Напишите программу, которая в последовательности натуральных десятичных чисел определяет среднее арифметическое элементов, запись которых в системе счисления с основанием 5 оканчивается **нечётной** цифрой. Если среди входных данных таких элементов нет, программа должна вывести «NO».

Программа получает на вход в первой строке натуральное число – количество чисел N ($3 \leq N \leq 10\,000$), затем N натуральных чисел, не превышающих 30 000, каждое в отдельной строке.

Программа должна вывести одно число – среднее арифметическое десятичных чисел (элементов последовательности), запись которых в 5-ричной системе счисления оканчивается **нечётной** цифрой, или «NO», если среди входных данных таких элементов нет.

Пример работы программы

Входные данные	Выходные данные
6 12 15 10 71 50 11	41

Номер: 2130E9

Напишите программу, которая в последовательности натуральных десятичных чисел определяет **наименьший** элемент, запись которого в системе счисления с основанием 9 оканчивается цифрой 1. Если среди входных данных таких элементов нет, программа должна вывести «NO».

Программа получает на вход в первой строке натуральное число – количество чисел N ($3 \leq N \leq 10\,000$), затем N натуральных чисел, не превышающих 30 000, каждое в отдельной строке.

Программа должна вывести одно десятичное число – **наименьший** элемент последовательности, запись которого в 9-ричной системе счисления оканчивается цифрой 1, или «NO», если среди входных данных таких элементов нет.

Пример работы программы

Входные данные	Выходные данные
5 10 9 11 91 28	10

Номер: D9A7E1

Напишите программу, которая в последовательности натуральных десятичных чисел определяет наибольший элемент, запись которого в системе счисления с основанием 9 оканчивается цифрой 1. Если среди входных данных таких элементов нет, программа должна вывести «NO».

Программа получает на вход натуральные десятичные числа, не превышающие 30 000, каждое в отдельной строке. Количество введенных чисел неизвестно, их последовательность заканчивается числом 0 (0 – признак окончания ввода, в последовательность не входит).

Программа должна вывести одно десятичное число – наибольший элемент последовательности, запись которого в 9-ричной системе счисления оканчивается цифрой 1, или «NO», если среди входных данных таких элементов нет.

Пример работы программы

Входные данные	Выходные данные
10	91
9	
11	
91	
28	
0	

Номер: 8A66E0

Напишите программу, которая в последовательности натуральных десятичных чисел определяет сумму элементов, запись которых в системе счисления с основанием 7 оканчивается цифрой 3. Если среди входных данных таких элементов нет, программа должна вывести «NO».

Программа получает на вход в первой строке натуральное число – количество чисел N ($3 \leq N \leq 10\,000$), затем N натуральных чисел, не превышающих 30 000, каждое в отдельной строке.

Программа должна вывести одно число – сумму десятичных чисел (элементов последовательности), запись которых в 7-ричной системе счисления оканчивается цифрой 3, или «NO», если среди входных данных таких элементов нет.

Пример работы программы

Входные данные	Выходные данные
5	41
33	
63	
17	
24	
53	

Номер: 46CC66

Напишите программу, которая в последовательности натуральных десятичных чисел определяет количество элементов, запись которых в системе счисления с основанием 7 оканчивается цифрой 2. Если среди входных данных таких элементов нет, программа должна вывести «NO».

Программа получает на вход натуральные десятичные числа, не превышающие 30 000, каждое в отдельной строке. Количество введённых чисел неизвестно, их последовательность заканчивается числом 0 (0 – признак окончания ввода, в последовательность не входит).

Программа должна вывести одно число – количество десятичных чисел (элементов последовательности), запись которых в 7-ричной системе счисления оканчивается цифрой 2, или «NO», если среди входных данных таких элементов нет.

Пример работы программы

Входные данные	Выходные данные
37 22 16 25 42 0	2

Номер: 1DF767

Напишите программу, которая в последовательности натуральных десятичных чисел определяет количество элементов, запись которых в системе счисления с основанием 5 оканчивается чётной цифрой. Если среди входных данных таких элементов нет, программа должна вывести «NO».

Программа получает на вход в первой строке натуральное число – количество чисел N ($3 \leq N \leq 10\,000$), затем N натуральных чисел, не превышающих 30 000, каждое в отдельной строке.

Программа должна вывести одно число – количество десятичных чисел (элементов последовательности), запись которых в 5-ричной системе счисления оканчивается чётной цифрой, или «NO», если среди входных данных таких элементов нет.

Пример работы программы

Входные данные	Выходные данные
6 12 15 10 71 50 11	4

Номер: 5A3C65

Напишите программу, которая в последовательности натуральных десятичных чисел определяет сумму элементов, запись которых в системе счисления с основанием 5 оканчивается **нечётной** цифрой. Если среди входных данных таких элементов нет, программа должна вывести «NO».

Программа получает на вход в первой строке натуральное число – количество чисел N ($3 \leq N \leq 10\,000$), затем N натуральных чисел, не превышающих 30 000, каждое в отдельной строке.

Программа должна вывести одно число – сумму десятичных чисел (элементов последовательности), запись которых в 5-ричной системе счисления оканчивается **нечётной** цифрой, или «NO», если среди входных данных таких элементов нет.

Пример работы программы

Входные данные	Выходные данные
6 12 15 10 71 50 11	82

Номер: 90CE3C

Напишите программу, которая в последовательности натуральных десятичных чисел определяет количество элементов, запись которых в системе счисления с основанием 5 оканчивается чётной цифрой. Если среди входных данных таких элементов нет, программа должна вывести «NO».

Программа получает на вход натуральные десятичные числа, не превышающие 30 000, каждое в отдельной строке. Количество введённых чисел неизвестно, их последовательность заканчивается числом 0 (0 – признак окончания ввода, в последовательность не входит).

Программа должна вывести одно число – количество десятичных чисел (элементов последовательности), запись которых в системе счисления с основанием 5 оканчивается чётной цифрой, или «NO», если среди входных данных таких элементов нет.

Пример работы программы

Входные данные	Выходные данные
12 15 10 71 50 11 0	4

Номер: EF2331

Напишите программу, которая в последовательности натуральных десятичных чисел определяет сумму элементов, запись которых в системе счисления с основанием 5 оканчивается **нечётной** цифрой. Если среди входных данных таких элементов нет, программа должна вывести «NO».

Программа получает на вход натуральные десятичные числа, не превышающие 30 000, каждое в отдельной строке. Количество введённых чисел неизвестно, их последовательность заканчивается числом 0 (0 – признак окончания ввода, в последовательность не входит).

Программа должна вывести одно число – сумму десятичных чисел (элементов последовательности), запись которых в системе счисления с основанием 5 оканчивается **нечётной** цифрой, или «NO», если среди входных данных таких элементов нет.

Пример работы программы

Входные данные	Выходные данные
12	82
15	
10	
71	
50	
11	
0	

Номер: 6D7939

Напишите программу, которая в последовательности натуральных десятичных чисел определяет **наименьший** элемент, запись которого в системе счисления с основанием 9 оканчивается цифрой 1. Если среди входных данных таких элементов нет, программа должна вывести «NO».

Программа получает на вход натуральные десятичные числа, не превышающие 30 000, каждое в отдельной строке. Количество введённых чисел неизвестно, их последовательность заканчивается числом 0 (0 – признак окончания ввода, в последовательность не входит).

Программа должна вывести одно десятичное число – **наименьший** элемент последовательности, запись которого в 9-ричной системе счисления оканчивается цифрой 1, или «NO», если среди входных данных таких элементов нет.

Пример работы программы

Входные данные	Выходные данные
10	10
9	
11	
91	
28	
0	

Номер: B8CB8B

Напишите программу, которая в последовательности натуральных десятичных чисел определяет наибольший элемент, запись которого в системе счисления с основанием 9 оканчивается цифрой 1. Если среди входных данных таких элементов нет, программа должна вывести «NO».

Программа получает на вход в первой строке натуральное число – количество чисел N ($3 \leq N \leq 10\,000$), затем N натуральных чисел, не превышающих 30 000, каждое в отдельной строке.

Программа должна вывести одно десятичное число – наибольший элемент последовательности, запись которого в 9-ричной системе счисления оканчивается цифрой 1, или «NO», если среди входных данных таких элементов нет.

Пример работы программы

Входные данные	Выходные данные
5	91
10	
9	
11	
91	
28	

Номер: 18EA84

Напишите программу, которая в последовательности натуральных десятичных чисел определяет сумму элементов, запись которых в системе счисления с основанием 7 оканчивается цифрой 6. Если среди входных данных таких элементов нет, программа должна вывести «NO».

Программа получает на вход в первой строке натуральное число – количество чисел N ($3 \leq N \leq 10\,000$), затем N натуральных чисел, не превышающих 30 000, каждое в отдельной строке.

Программа должна вывести одно число – сумму десятичных чисел (элементов последовательности), запись которых в 7-ричной системе счисления оканчивается цифрой 6, или «NO», если среди входных данных таких элементов нет.

Пример работы программы

Входные данные	Выходные данные
5	33
20	
13	
9	
28	
12	

Номер: 1B108F

Напишите программу, которая в последовательности натуральных десятичных чисел определяет среднее арифметическое элементов, запись которых в си-

стеме счисления с основанием 7 оканчивается цифрой 4. Если среди входных данных таких элементов нет, программа должна вывести «NO».

Программа получает на вход натуральные десятичные числа, не превышающие 30 000, каждое в отдельной строке. Количество введённых чисел неизвестно, их последовательность заканчивается числом 0 (0 – признак окончания ввода, в последовательность не входит).

Программа должна вывести одно число – среднее арифметическое десятичных чисел (элементов последовательности), запись которых в 7-ричной системе счисления оканчивается цифрой 4, или «NO», если среди входных данных таких элементов нет.

Пример работы программы

Входные данные	Выходные данные
11	18
25	
30	
14	
54	
0	

Номер: B8D5FC

Напишите программу, которая в последовательности натуральных чисел определяет сумму чисел, кратных 3 и оканчивающихся на 8. Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число, кратное 3 и оканчивающееся на 8.

Количество чисел не превышает 100. Введённые числа по модулю не превышают 300.

Программа должна вывести одно число: сумму чисел, кратных 3 и оканчивающихся на 8.

Пример работы программы:

Входные данные	Выходные данные
3 18 25 48	66

Номер: 17BAFA

Напишите программу, которая в последовательности натуральных чисел определяет сумму чисел, кратных 5. Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число, кратное 5.

Количество чисел не превышает 100. Введённые числа не превышают 300. Программа должна вывести одно число – сумму чисел, кратных 5.

Пример работы программы:

Входные данные	Выходные данные
3 15 25 6	40

Номер: D7DDF7

Напишите программу, которая в последовательности натуральных чисел находит среднее арифметическое чисел, кратных 8, или сообщает, что таких чисел нет (выводит «NO»). Программа получает на вход натуральные числа, количество введённых чисел неизвестно, последовательность чисел заканчивается числом 0 (0 – признак окончания ввода, не входит в последовательность).

Количество чисел не превышает 100. Введённые числа не превышают 300. Программа должна вывести среднее арифметическое чисел, кратных 8, или вывести «NO», если таких чисел нет. Значение выводить с точностью до десятых.

Пример работы программы:

Входные данные	Выходные данные
8 122 64 16 0	29.3
111 1 0	NO

Номер: A367FD

Напишите программу для решения следующей задачи.

Ученики 4 класса вели дневники наблюдения за погодой и ежедневно записывали дневную температуру. Найдите среднюю температуру за время наблюдения. Если количество дней, когда температура поднималась выше нуля градусов, не менее 5, выведите YES, иначе выведите NO.

Программа получает на вход количество дней, в течение которых проводилось наблюдение N ($1 \leq N \leq 31$), затем для каждого дня вводится температура.

Пример работы программы:

Входные данные	Выходные данные
4 —4 12 —2 8	3.5 NO

Номер: C576FE

Напишите программу, которая в последовательности целых чисел определяет количество чисел, кратных 5 или 9. Программа получает на вход целые числа, количество введённых чисел неизвестно, последовательность чисел заканчивается числом 0 (0 – признак окончания ввода, не входит в последовательность).

Количество чисел не превышает 1000. Введённые числа по модулю не превышают 30 000.

Программа должна вывести одно число: количество чисел, кратных 5 или 9.

Пример работы программы:

Входные данные	Выходные данные
15 91 90 0	2

Номер: 3F4FFe

Напишите программу, которая в последовательности целых чисел определяет количество чётных чисел, кратных 5. Программа получает на вход целые числа, количество введённых чисел неизвестно, последовательность чисел заканчивается числом 0 (0 – признак окончания ввода, не входит в последовательность).

Количество чисел не превышает 1000. Введённые числа по модулю не превышают 30 000. Программа должна вывести одно число: количество чётных чисел, кратных 5.

Пример работы программы:

Входные данные	Выходные данные
—32 10 17 0	1

Номер: 17F601

Напишите программу, которая в последовательности натуральных чисел вычисляет сумму трёхзначных чисел, кратных 4. Программа получает на вход натуральные числа, количество введённых чисел неизвестно, последовательность чисел заканчивается числом 0 (0 – признак окончания ввода, не входит в последовательность).

Количество чисел не превышает 20. Введённые числа не превышают 1500. Программа должна вывести одно число: сумму трёхзначных чисел, кратных 4.

Пример работы программы:

Входные данные	Выходные данные
18	296
192	
104	
117	
0	

Номер: CACD02

Напишите программу, которая в последовательности натуральных чисел определяет количество однозначных чисел, кратных 3. Программа получает на вход натуральные числа, количество введённых чисел неизвестно, последовательность чисел заканчивается числом 0 (0 – признак окончания ввода, не входит в последовательность).

Количество чисел не превышает 1000. Введённые числа не превышают 30 000. Программа должна вывести одно число: количество однозначных чисел, кратных 3.

Пример работы программы:

Входные данные	Выходные данные
121 9 4 0	1

Номер: 37BA0C

Напишите программу, которая в последовательности натуральных чисел определяет минимальное число, оканчивающееся на 6. Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число, оканчивающееся на 6.

Количество чисел не превышает 1000. Введённые числа не превышают 30 000. Программа должна вывести одно число – минимальное число, оканчивающееся на 6.

Пример работы программы:

Входные данные	Выходные данные
3 26 16 36	16

Номер: 4A457B

Напишите программу, которая в последовательности натуральных чисел определяет минимальное число, оканчивающееся на 2. Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число, оканчивающееся на 2.

Количество чисел не превышает 1000. Введённые числа по модулю не превышают 30 000.

Программа должна вывести одно число: минимальное число, оканчивающееся на 2.

Пример работы программы:

Входные данные	Выходные данные
3 22 12 36	12

Номер: 5A5C7A

Напишите программу для решения следующей задачи.

На контрольной работе по алгебре ученикам 9 класса было предложено 10 примеров. Неудовлетворительная оценка выставляется, если правильно решено менее половины примеров. Сколько неудовлетворительных оценок было получено учениками? Если хотя бы один из учеников правильно решил все задачи, выведите YES, иначе выведите NO.

Программа получает на вход количество учеников в классе N ($1 \leq N \leq 30$), затем для каждого ученика вводится количество правильно решённых примеров.

Пример работы программы:

Входные данные	Выходные данные
4 3 9 2 8	2 NO

Номер: EE667E

Напишите программу, которая в последовательности натуральных чисел определяет количество чисел, кратных 6. Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число, кратное 6.

Количество чисел не превышает 1000. Введённые числа не превышают 30 000. Программа должна вывести одно число – количество чисел, кратных 6.

Пример работы программы:

Входные данные	Выходные данные
3 18 26 24	2

Номер: 689C75

Напишите программу, которая в последовательности натуральных чисел находит среднее арифметическое двузначных чисел или сообщает, что таких чисел нет (выводит «NO»). Программа получает на вход натуральные числа, количество введённых чисел неизвестно, последовательность чисел заканчивается числом 0 (0 – признак окончания ввода, не входит в последовательность).

Количество чисел не превышает 100. Введенные числа не превышают 300.

Программа должна вывести среднее арифметическое двузначных чисел или вывести «NO», если таких чисел нет. Значение выводить с точностью до десятых.

Пример работы программы:

Входные данные	Выходные данные
10 120 49 0	29.5
111 1 0	NO

Номер: e61278

Напишите программу, которая в последовательности натуральных чисел определяет количество чисел, оканчивающихся на 9. Программа получает на

вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число, оканчивающееся на 9.

Количество чисел не превышает 1000. Введённые числа по модулю не превышают 30 000.

Программа должна вывести одно число: количество чисел, оканчивающихся на 9.

Пример работы программы:

Входные данные	Выходные данные
3 19 29 24	2

Номер: 18ВЕВ8

Напишите программу, которая в последовательности натуральных чисел определяет сумму чисел, кратных 3 и оканчивающихся на 4. Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число, кратное 3 и оканчивающееся на 4.

Количество чисел не превышает 100. Введённые числа по модулю не превышают 300.

Программа должна вывести одно число: сумму чисел, кратных 3 и оканчивающихся на 4.

Пример работы программы:

Входные данные	Выходные данные
3 24 25 54	78

Номер: С3АЕВ5

Напишите программу, которая по двум данным натуральным числам a и b , не превосходящим 30000, подсчитывает количество чётных натуральных чисел на отрезке $[a, b]$ (включая концы отрезка).

Программа получает на вход два натуральных числа a и b , при этом гарантируется, что $1 \leq a \leq b \leq 30000$. Проверять входные данные на корректность не нужно.

Программа должна вывести одно число: количество чётных чисел на отрезке $[a, b]$.

Пример работы программы:

Входные данные	Выходные данные
10 20	6

Номер: E2АЕВ9

Напишите программу, которая в последовательности натуральных чисел определяет минимальное число, оканчивающееся на 4. Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число, оканчивающееся на 4.

Количество чисел не превышает 1000. Введённые числа не превышают 30 000.

Программа должна вывести одно число – минимальное число, оканчивающееся на 4.

Пример работы программы:

Входные данные	Выходные данные
3 24 14 34	14

Номер: F66F1D

Напишите программу, которая в последовательности натуральных чисел определяет количество чисел, кратных 3 и оканчивающихся на 2. Программа получает на вход количество чисел в последовательности, а затем сами числа.

Количество чисел не превышает 1000. Введённые числа по модулю не превышают 30 000.

Программа должна вывести одно число: количество чисел, кратных 3 и оканчивающихся на 2.

Пример работы программы:

Входные данные	Выходные данные
4 12 25 12 9	2

Номер: DE6B1E

Напишите программу, которая в последовательности натуральных чисел определяет количество чисел, кратных 3. Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число, кратное 3.

Количество чисел не превышает 1000. Введённые числа не превышают 30 000.

Программа должна вывести одно число – количество чисел, кратных 3.

Пример работы программы:

Входные данные	Выходные данные
3 12 26 24	2

Номер: 51BD18

Напишите программу, которая в последовательности натуральных чисел определяет количество чисел, оканчивающихся на 8. Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число, оканчивающееся на 8.

Количество чисел не превышает 1000. Введённые числа по модулю не превышают 30 000.

Программа должна вывести одно число: количество чисел, оканчивающихся на 8.

Пример работы программы:

Входные данные	Выходные данные
3 18 28 24	2

Номер: A55719

Напишите программу, которая в последовательности натуральных чисел определяет сумму чисел, кратных 6 и оканчивающихся на 8. Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число, кратное 6 и оканчивающееся на 8.

Количество чисел не превышает 100. Введённые числа по модулю не превышают 300.

Программа должна вывести одно число: сумму чисел, кратных 6 и оканчивающихся на 8.

Пример работы программы:

Входные данные	Выходные данные
3 18 25 48	66

Номер: EB771C

Напишите программу, которая в последовательности натуральных чисел определяет сумму чисел, оканчивающихся на 4. Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число, оканчивающееся на 4.

Количество чисел не превышает 100. Введённые числа не превышают 300.

Программа должна вывести одно число – сумму чисел, оканчивающихся на 4.

Пример работы программы:

Входные данные	Выходные данные
3 14 25 24	38

Номер: 73582D

Напишите программу, которая в последовательности натуральных чисел определяет количество чисел, оканчивающихся на 3. Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число, оканчивающееся на 3.

Количество чисел не превышает 1000. Введённые числа не превышают 30 000.

Программа должна вывести одно число – количество чисел, оканчивающихся на 3.

Пример работы программы:

Входные данные	Выходные данные
3 13 23 24	2

Номер: V8BD27

Напишите программу, которая в последовательности натуральных чисел определяет максимальное число, кратное 8. Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число, кратное 8.

Количество чисел не превышает 1000. Введённые числа по модулю не превышают 30 000.

Программа должна вывести одно число: максимальное число, кратное 8.

Пример работы программы:

Входные данные	Выходные данные
3 16 24 12	24

Номер: V92B25

Напишите программу, которая в последовательности натуральных чисел определяет сумму чисел, оканчивающихся на 8. Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число, оканчивающееся на 8.

Количество чисел не превышает 100. Введённые числа по модулю не превышают 300.

Программа должна вывести одно число: сумму чисел, оканчивающихся на 8.

Пример работы программы:

Входные данные	Выходные данные
3 18 28 24	46

Номер: 2DE92F

Напишите программу, которая в последовательности натуральных чисел вычисляет сумму всех однозначных чисел, кратных 3. Программа получает на вход натуральные числа, количество введённых чисел неизвестно, последовательность чисел заканчивается числом 0 (0 – признак окончания ввода, не входит в последовательность).

Количество чисел не превышает 100. Введённые числа не превышают 300.

Программа должна вывести одно число: сумму всех однозначных чисел, кратных 3.

Пример работы программы:

Входные данные	Выходные данные
6 6 17 18 0	12

Номер: 2есВ20

Напишите программу, которая в последовательности целых чисел определяет количество чётных чисел, кратных 9. Программа получает на вход целые числа, количество введённых чисел неизвестно, последовательность чисел заканчивается числом 0 (0 – признак окончания ввода, не входит в последовательность).

Количество чисел не превышает 1000. Введённые числа по модулю не превышают 30 000.

Программа должна вывести одно число: количество чётных чисел, кратных 9.

Пример работы программы:

Входные данные	Выходные данные
-32 18 17 0	1

Номер: 2АС626

Напишите программу, которая в последовательности натуральных чисел определяет количество чисел, кратных 6 и оканчивающихся на 2. Программа получает на вход количество чисел в последовательности, а затем сами числа.

Количество чисел не превышает 1000. Введённые числа по модулю не превышают 30 000.

Программа должна вывести одно число: количество чисел, кратных 6 и оканчивающихся на 2.

Пример работы программы:

Входные данные	Выходные данные
3 12 25 42	2

Номер: 50F022

Напишите программу, которая в последовательности натуральных чисел определяет сумму чисел, кратных 3. Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число, кратное 3.

Количество чисел не превышает 100. Введённые числа не превышают 300.

Программа должна вывести одно число – сумму чисел, кратных 3.

Пример работы программы:

Входные данные	Выходные данные
3 12 25 9	21

Номер: АЕ0327

Напишите программу, которая в последовательности натуральных чисел вычисляет сумму всех чисел, кратных 7 и оканчивающихся на 0. Программа полу-

чает на вход натуральные числа, количество введённых чисел неизвестно, последовательность чисел заканчивается числом 0 (0 – признак окончания ввода, не входит в последовательность).

Количество чисел не превышает 100. Введённые числа не превышают 300.

Программа должна вывести одно число: сумму всех чисел, кратных 7 и оканчивающихся на 0.

Пример работы программы:

Входные данные	Выходные данные
14 140 20 70 0	210

Номер: A0B42B

Напишите программу, которая в последовательности натуральных чисел определяет количество двузначных чисел, кратных 8. Программа получает на вход натуральные числа, количество введённых чисел неизвестно, последовательность чисел заканчивается числом 0 (0 – признак окончания ввода, не входит в последовательность).

Количество чисел не превышает 1000. Введённые числа не превышают 30 000.

Программа должна вывести одно число: количество двузначных чисел, кратных 8.

Пример работы программы:

Входные данные	Выходные данные
8 16 77 0	1

Номер: 937E26

Напишите программу, которая в последовательности натуральных чисел находит среднее арифметическое трёхзначных чисел или сообщает, что таких чисел нет (выводит NO). Программа получает на вход натуральные числа, количество введённых чисел неизвестно, последовательность чисел заканчивается числом 0 (0 – признак окончания ввода, не входит в последовательность).

Количество чисел не превышает 20. Введённые числа не превышают 1500.

Программа должна вывести среднее арифметическое трёхзначных чисел или вывести «NO», если таких чисел нет.

Пример работы программы:

Входные данные	Выходные данные
10 120 125 0	122.5

11 1 0	NO
--------------	----

Номер: 40A3D0

Напишите программу, которая в последовательности целых чисел определяет количество чисел, кратных 6 или 11. Программа получает на вход целые числа, количество введённых чисел неизвестно, последовательность чисел заканчивается числом 0 (0 – признак окончания ввода, не входит в последовательность).

Количество чисел не превышает 1000. Введённые числа не превышают 30 000.

Программа должна вывести одно число: количество чисел, кратных 6 или 11.

Пример работы программы:

Входные данные	Выходные данные
66 91 41 60 0	2

Номер: 4DD2D2

Напишите программу, которая в последовательности натуральных чисел определяет минимальное число, кратное 9. Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число, кратное 9.

Количество чисел не превышает 1000. Введённые числа по модулю не превышают 30 000.

Программа должна вывести одно число: минимальное число, кратное 9.

Пример работы программы:

Входные данные	Выходные данные
3 18 9 31	9

Номер: 498AD5

Напишите программу, которая в последовательности натуральных чисел определяет количество чисел, кратных 3 и оканчивающихся на 4. Программа получает на вход количество чисел в последовательности, а затем сами числа.

Количество чисел не превышает 1000. Введённые числа по модулю не превышают 30 000.

Программа должна вывести одно число: количество чисел, кратных 3 и оканчивающихся на 4.

Пример работы программы:

Входные данные	Выходные данные
3 24 25 54	2

Номер: 08A5D8

Напишите программу, которая в последовательности целых чисел вычисляет сумму всех чисел, которые кратны 4 или 9. Программа получает на вход целые числа, количество введённых чисел неизвестно, последовательность чисел заканчивается числом 0 (0 – признак окончания ввода, не входит в последовательность).

Количество чисел не превышает 20. Введённые числа по модулю не превышают 1500.

Программа должна вывести одно число: сумму всех чисел, которые кратны 4 или 9.

Пример работы программы:

Входные данные	Выходные данные
-16 93 90 0	74

Номер: 1C81DB

Напишите программу, которая в последовательности целых чисел определяет количество чисел, кратных 7 и оканчивающихся на ноль. Программа получает на вход целые числа, количество введённых чисел неизвестно, последовательность чисел заканчивается числом 0 (0 – признак окончания ввода, не входит в последовательность).

Количество чисел не превышает 1000. Введённые числа по модулю не превышают 30 000.

Программа должна вывести одно число: количество чисел, кратных 7 и оканчивающихся на ноль.

Пример работы программы:

Входные данные	Выходные данные
27 140 15 0	1

Номер: 212EDE

Напишите программу, которая в последовательности натуральных чисел определяет количество чисел, кратных 4. Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число, кратное 4.

Количество чисел не превышает 1000. Введённые числа не превышают 30 000.

Программа должна вывести одно число – количество чисел, кратных 4.

Пример работы программы:

Входные данные	Выходные данные
3 16 26 24	2

Номер: AFEFD1

Напишите программу, которая в последовательности целых чисел вычисляет сумму всех чисел, которые кратны 6 или 11. Программа получает на вход целые числа, количество введённых чисел неизвестно, последовательность чисел заканчивается числом 0 (0 – признак окончания ввода, не входит в последовательность).

Количество чисел не превышает 100. Введённые числа по модулю не превышают 300.

Программа должна вывести одно число: сумму всех чисел, которые кратны 6 или 11.

Пример работы программы:

Входные данные	Выходные данные
12 11 8 0	23

Номер: C681D5

Напишите программу для решения следующей задачи.

Ученики 4 класса вели дневники наблюдения за погодой и ежедневно записывали дневную температуру. Найдите самую низкую температуру за время наблюдения. Если температура опускалась ниже -15 градусов, выведите YES, иначе выведите NO.

Программа получает на вход количество дней, в течение которых проводилось измерение температуры N ($1 \leq N \leq 31$), затем для каждого дня вводится температура.

Пример работы программы:

Входные данные	Выходные данные
4 -5 12 -2 8	-5 NO

Номер: 8E46D4

Напишите программу для решения следующей задачи.

Ученики 4 класса вели дневники наблюдения за погодой и ежедневно записывали дневную температуру. Найдите среднюю температуру для дней, когда температура поднималась выше нуля градусов. Определите количество таких дней. Гарантируется, что за время наблюдения хотя бы в один из дней температура поднималась выше нуля градусов.

Программа получает на вход количество дней, в течение которых проводилось наблюдение N ($1 \leq N \leq 31$), затем для каждого дня вводится температура.

Пример работы программы:

Входные данные	Выходные данные
4 -5 12 -2 8	10.0 2

Номер: 415E52,

Напишите программу, которая в последовательности натуральных чисел определяет минимальное чётное число.

Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется чётное число.

Количество чисел не превышает 1000. Введённые числа не превышают 30 000.

Программа должна вывести одно число – минимальное чётное число.

Пример работы программы:

Входные данные	Выходные данные
4 3 20 6 8	6

Номер: 51A753

Напишите программу, которая в последовательности натуральных чисел определяет количество чисел, кратных 3 и оканчивающихся на 8. Программа получает на вход количество чисел в последовательности, а затем сами числа.

Количество чисел не превышает 1000. Введённые числа по модулю не превышают 30 000.

Программа должна вывести одно число: количество чисел, кратных 3 и оканчивающихся на 8.

Пример работы программы:

Входные данные	Выходные данные
3 18 25 48	2

Номер: 4E7CA2

Напишите программу, которая в последовательности целых чисел находит среднее арифметическое чисел, кратных 9, или сообщает, что таких чисел нет. Программа получает на вход целые числа, количество введённых чисел неизвестно, последовательность чисел заканчивается числом 0 (0 – признак окончания ввода, не является членом последовательности).

Количество чисел не превышает 100. Введённые числа по модулю не превышают 300.

Программа должна вывести среднее арифметическое чисел, кратных 9, или вывести NO, если таких чисел в последовательности нет.

Пример работы программы:

Входные данные	Выходные данные
9 -30 18 0	13.5
-15 7 0	NO

Номер: 151BA0

Напишите программу, которая в последовательности натуральных чисел находит наименьшее число, кратное 3. Программа получает на вход целые числа, количество введённых чисел неизвестно, последовательность чисел заканчивается числом 0 (0 – признак окончания ввода, не является членом последовательности).

Количество чисел не превышает 1000. Введённые числа не превышают 30000. Гарантируется, что в последовательности есть хотя бы одно число, кратное 3.

Программа должна вывести наименьшее число, кратное 3.

Пример работы программы:

Входные данные	Выходные данные
45 10 9 0	9

Номер: AEEAA5

Напишите программу, которая в последовательности натуральных чисел определяет максимальное число, кратное 5. Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число, кратное 5.

Количество чисел не превышает 1000. Введённые числа не превышают 30 000. Программа должна вывести одно число – максимальное число, кратное 5.

Пример работы программы:

Входные данные	Выходные данные
3 10 25 12	25

Номер: 8D33AE

Напишите программу, которая в последовательности натуральных чисел находит среднее арифметическое чисел, кратных 4, или сообщает, что таких чисел

нет (выводит NO). Программа получает на вход натуральные числа, количество введённых чисел неизвестно, последовательность чисел заканчивается числом 0 (0 – признак окончания ввода, не входит в последовательность).

Количество чисел не превышает 100. Введённые числа не превышают 300.

Программа должна вывести среднее арифметическое чисел, кратных 4 или вывести «NO», если таких чисел нет.

Пример работы программы:

Входные данные	Выходные данные
8 120 64 16 0	52.0
111 1 0	NO

Номер: 45C8CD

Напишите программу, которая в последовательности натуральных чисел определяет количество чисел, кратных 6 и оканчивающихся на 8. Программа получает на вход количество чисел в последовательности, а затем сами числа.

Количество чисел не превышает 1000. Введённые числа по модулю не превышают 30 000.

Программа должна вывести одно число: количество чисел, кратных 6 и оканчивающихся на 8.

Пример работы программы:

Входные данные	Выходные данные
3 18 25 48	2

Номер: 2C42CA

Напишите программу, которая в последовательности натуральных чисел определяет сумму чисел, кратных 6 и оканчивающихся на 4. Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число, кратное 6 и оканчивающееся на 4.

Количество чисел не превышает 100. Введённые числа по модулю не превышают 300.

Программа должна вывести одно число: сумму чисел, кратных 6 и оканчивающихся на 4.

Пример работы программы:

Входные данные	Выходные данные
3 24 25 54	78

Номер: DF90CE

Напишите программу, которая в последовательности натуральных чисел определяет сумму чисел, кратных 6. Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число, кратное 6.

Количество чисел не превышает 100. Введённые числа не превышают 300.

Программа должна вывести одно число — сумму чисел, кратных 6.

Пример работы программы:

Входные данные	Выходные данные
3 12 25 6	18

Номер: 30E5C9

Напишите программу, которая в последовательности целых чисел находит сумму и количество чисел, кратных 17, или сообщает, что таких чисел нет. На вход программы подаются целые числа, количество введённых чисел неизвестно, последовательность заканчивается числом 0 (0 – признак окончания ввода, не является членом последовательности).

Количество чисел не превышает 100. Введённые числа по модулю не превышают 300.

Программа должна вывести сумму и количество чисел, кратных 17, или вывести NO, если таких чисел в последовательности нет.

Пример работы программы:

Входные данные	Выходные данные
34 30 17 0	51 2
-16 5 0	NO

Номер: 2E2591

Напишите программу, которая в последовательности натуральных чисел находит среднее арифметическое двузначных чисел или сообщает, что таких чисел нет. Программа получает на вход натуральные числа, количество введённых чисел неизвестно, последовательность заканчивается числом 0 (0 – признак окончания ввода, не является членом последовательности).

Количество чисел не превышает 100. Введённые числа не превышают 300.

Программа должна вывести среднее арифметическое двузначных чисел или вывести NO, если таких чисел в последовательности нет.

Пример работы программы:

Входные данные	Выходные данные
13 154 56 0	34.5
269 5 0	NO

Номер: 59429F

Напишите программу для решения следующей задачи.

Участники парусной регаты стартовали одновременно. На финише фиксировалось время прохождения маршрута каждой яхтой (в часах и минутах). Определите время победителя регаты (в часах и минутах). Известно, что соревнования проходили в течение 12 часов.

Программа получает на вход количество яхт, принимавших участие в регате N ($1 \leq N \leq 100$), затем для каждой яхты вводится два числа: часы и минуты, затраченные на прохождение маршрута.

Пример работы программы:

Входные данные	Выходные данные
2 3 25 2 50	2 50

Номер: AA7391

Напишите программу, которая в последовательности натуральных чисел вычисляет сумму всех двузначных чисел, кратных 8. Программа получает на вход натуральные числа, количество введённых чисел неизвестно, последовательность чисел заканчивается числом 0 (0 – признак окончания ввода, не входит в последовательность).

Количество чисел не превышает 20. Введённые числа не превышают 1500.

Программа должна вывести одно число: сумму всех двузначных чисел, кратных 8.

Пример работы программы:

Входные данные	Выходные данные
17 16 32 160 0	48

Номер: A07498

Напишите программу, которая в последовательности целых чисел определяет количество чисел, кратных 5 или 7. Программа получает на вход целые числа, количество введённых чисел неизвестно, последовательность чисел заканчивается числом 0 (0 – признак окончания ввода, не входит в последовательность).

Количество чисел не превышает 1000. Введённые числа по модулю не превышают 30 000.

Программа должна вывести одно число: количество чисел, кратных 5 или 7.

Пример работы программы:

Входные данные	Выходные данные
-22 25 -14 0	2

Номер: CAF691

Напишите программу, которая в последовательности натуральных чисел определяет сумму чисел, кратных 3 и оканчивающихся на 2. Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число, кратное 3 и оканчивающееся на 2.

Количество чисел не превышает 100. Введённые числа по модулю не превышают 300.

Программа должна вывести одно число: сумму чисел, кратных 3 и оканчивающихся на 2.

Пример работы программы:

Входные данные	Выходные данные
4 12 25 12 9	24

Номер: 41E9E1

Напишите программу для решения следующей задачи.

Девятиклассники участвовали в викторине по математике. Необходимо было ответить на 20 вопросов. Победителем викторины считается участник, правильно ответивший на наибольшее количество вопросов. На сколько вопросов победитель ответил правильно? Если есть участники викторины, которые не смогли дать правильный ответ ни на один из вопросов, выведите YES, иначе выведите NO. Гарантируется, что есть участники, правильно ответившие хотя бы на один из вопросов.

Программа получает на вход число участников викторины N ($1 \leq N \leq 50$), затем для каждого участника вводится количество вопросов, на которые получен правильный ответ.

Пример работы программы:

Входные данные	Выходные данные
4 15 12 0 17	17 YES

Номер: ВА6ВЕ0

Напишите программу, которая в последовательности натуральных чисел определяет сумму чисел, кратных 6 и оканчивающихся на 2. Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число, кратное 6 и оканчивающееся на 2.

Количество чисел не превышает 100. Введённые числа по модулю не превышают 300.

Программа должна вывести одно число: сумму чисел, кратных 6 и оканчивающихся на 2.

Пример работы программы:

Входные данные	Выходные данные
3 12 25 42	54

Номер: 20E8E3

Напишите программу, которая в последовательности натуральных чисел определяет сумму чисел, оканчивающихся на 3. Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число, оканчивающееся на 3.

Количество чисел не превышает 100. Введённые числа не превышают 300.

Программа должна вывести одно число — сумму чисел, оканчивающихся на 3.

Пример работы программы:

Входные данные	Выходные данные
3 13 23 24	36

Номер: DB34E2

Напишите программу для решения следующей задачи.

На соревнованиях по бегу на 5000 метров для выхода в финал необходимо показать время не более 18 минут 30 секунд.

Сколько спортсменов по результатам соревнований вышли в финал?

Программа получает на вход количество спортсменов, принимавших участие в соревнованиях ($1 \leq N \leq 100$), затем для каждого спортсмена вводится два числа: минуты и секунды – время, показанное на дистанции.

Пример работы программы:

Входные данные	Выходные данные
2 17 15 19 20	1

Номер: 516CE0

Напишите программу, которая в последовательности натуральных чисел определяет количество чисел, оканчивающихся на 6. Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число, оканчивающееся на 6.

Количество чисел не превышает 1000. Введённые числа не превышают 30 000.

Программа должна вывести одно число — количество чисел, оканчивающихся на 6.

Пример работы программы:

Входные данные	Выходные данные
3 16 26 24	2

Номер: C2D6EB

Напишите программу, которая в последовательности целых чисел вычисляет сумму всех чисел, кратных 5 или 9. Программа получает на вход целые числа, количество введённых чисел неизвестно, последовательность чисел заканчивается числом 0 (0 – признак окончания ввода, не входит в последовательность).

Количество чисел не превышает 20. Введённые числа по модулю не превышают 1500.

Программа должна вывести одно число: сумму всех чисел, кратных 5 или 9.

Пример работы программы:

Входные данные	Выходные данные
-15 99 8 0	84

Номер: 3CBFE3

Напишите программу, которая в последовательности натуральных чисел определяет максимальное число, оканчивающееся на 2.

Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число, оканчивающееся на 2.

Количество чисел не превышает 1000. Введённые числа не превышают 30 000.

Программа должна вывести одно число – максимальное число, оканчивающееся на 2.

Пример работы программы:

Входные данные	Выходные данные
4 3 22 6 12	22

Номер: 8D8BEF

Напишите программу, которая в последовательности натуральных чисел определяет количество чисел, кратных 6 и оканчивающихся на 4. Программа получает на вход натуральные числа, количество введённых чисел неизвестно, последовательность чисел заканчивается числом 0 (0 – признак окончания ввода, не входит в последовательность).

Количество чисел не превышает 1000. Введённые числа не превышают 30 000.

Программа должна вывести одно число: количество чисел, кратных 6 и оканчивающихся на 4.

Пример работы программы:

Входные данные	Выходные данные
12 24 22 0	1

Номер: 02A267

Напишите программу, которая в последовательности целых чисел определяет количество чётных чисел, кратных 7. Программа получает на вход целые числа, количество введённых чисел неизвестно, последовательность чисел заканчивается числом 0 (0 – признак окончания ввода, не входит в последовательность).

Количество чисел не превышает 1000. Введённые числа по модулю не превышают 30 000.

Программа должна вывести одно число: количество чётных чисел, кратных 7.

Пример работы программы:

Входные данные	Выходные данные
-32 14 17 0	1

Номер: 7AC466

Напишите программу, которая в последовательности натуральных чисел определяет сумму всех чисел, кратных 6 и оканчивающихся на 4. Программа получает на вход натуральные числа, количество введённых чисел неизвестно, последовательность чисел заканчивается числом 0 (0 – признак окончания ввода, не входит в последовательность).

Количество чисел не превышает 100. Введённые числа не превышают 300.

Программа должна вывести одно число: сумму всех чисел, кратных 6 и оканчивающихся на 4.

Пример работы программы:

Входные данные	Выходные данные
14 24 144 22 12 0	168

Номер: 3FED6B

Напишите программу, которая в последовательности натуральных чисел определяет максимальное число, оканчивающееся на 3. Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число, оканчивающееся на 3.

Количество чисел не превышает 1000. Введённые числа не превышают 30 000.

Программа должна вывести одно число – максимальное число, оканчивающееся на 3.

Пример работы программы:

Входные данные	Выходные данные
3 13 23 3	23

Номер: 819D6B

Напишите программу, которая в последовательности натуральных чисел определяет минимальное число, кратное 3. Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число, кратное 3.

Количество чисел не превышает 1000. Введённые числа не превышают 30 000. Программа должна вывести одно число — минимальное число, кратное 3.

Пример работы программы:

Входные данные	Выходные данные
3 21 12 31	12

Номер: 5B4731

Напишите программу, которая в последовательности натуральных чисел определяет количество чисел, кратных 6 и оканчивающихся на 4. Программа получает на вход количество чисел в последовательности, а затем сами числа.

Количество чисел не превышает 1000. Введённые числа по модулю не превышают 30 000.

Программа должна вывести одно число: количество чисел, кратных 6 и оканчивающихся на 4.

Пример работы программы:

Входные данные	Выходные данные
3 24 25 54	2

Номер: 87A333

Напишите программу для решения следующей задачи.

На зачётной работе по физике ученикам 9 класса было предложено 12 задач. Зачёт можно получить, если правильно решить не менее 8 задач. Сколько учеников получило зачёт? Вычислите среднее количество правильно решённых задач учащимися, получившими зачёт. Гарантируется, что хотя бы один ученик сдал зачёт.

Программа получает на вход количество учеников в классе N ($1 \leq N \leq 30$), затем для каждого ученика вводится количество правильно решённых задач.

Пример работы программы:

Входные данные	Выходные данные
4	2
3	9.5
9	
10	
6	

Номер: 49F582

Напишите программу, которая в последовательности натуральных чисел определяет количество трёхзначных чисел, кратных 4. Программа получает на вход натуральные числа, количество введённых чисел неизвестно, последовательность чисел заканчивается числом 0 (0 – признак окончания ввода, не входит в последовательность).

Количество чисел не превышает 1000. Введённые числа не превышают 30 000.

Программа должна вывести одно число: количество трёхзначных чисел, кратных 4.

Пример работы программы:

Входные данные	Выходные данные
120	1
9	
365	
4	
0	

Номер: 431786

Напишите программу, которая в последовательности натуральных чисел находит среднее арифметическое чисел, оканчивающихся на единицу, или сообща-

ет, что таких чисел нет. Программа получает на вход целые числа, количество введённых чисел неизвестно, последовательность чисел заканчивается числом 0 (0 – признак окончания ввода).

Количество чисел не превышает 100. Введённые числа по модулю не превышают 300.

Программа должна вывести среднее арифметическое чисел, оканчивающихся на единицу, или вывести NO, если таких чисел в последовательности нет.

Пример работы программы:

Входные данные	Выходные данные
21 30 11 0	16.0
16 5 0	NO

Номер: 0B9580

Напишите программу, которая в последовательности целых чисел определяет количество чисел, кратных 4 и оканчивающихся на 0. Программа получает на вход целые числа, количество введённых чисел неизвестно, последовательность чисел заканчивается числом 0 (0 – признак окончания ввода, не является членом последовательности).

Количество чисел не превышает 1000. Введённые числа по модулю не превышают 30 000.

Программа должна вывести одно число: количество чисел, кратных 4 и оканчивающихся на 0.

Пример работы программы:

Входные данные	Выходные данные
20 40 -27 12 0	2

Номер: 046587

Напишите программу, которая в последовательности натуральных чисел определяет максимальное число, кратное 4. Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число, кратное 4.

Количество чисел не превышает 1000. Введённые числа не превышают 30 000. Программа должна вывести одно число – максимальное число, кратное 4.

Пример работы программы:

Входные данные	Выходные данные
3 8 16 11	16

Номер: 7D2B86

Напишите программу, которая в последовательности целых чисел определяет количество чисел, кратных 4 или 9. Программа получает на вход целые числа, количество введённых чисел неизвестно, последовательность чисел заканчивается числом 0 (0 – признак окончания ввода, не входит в последовательность).

Количество чисел не превышает 1000. Введённые числа по модулю не превышают 30 000.

Программа должна вывести одно число: количество чисел, кратных 4 или 9.

Пример работы программы:

Входные данные	Выходные данные
36 91 40 0	2

Номер: 8C748F

Напишите программу, которая в последовательности натуральных чисел определяет количество таких чисел, которые кратны 4 и оканчиваются на 2. Программа получает на вход натуральные числа, количество введённых чисел неизвестно, последовательность чисел заканчивается числом 0 (0 – признак окончания ввода, не входит в последовательность).

Количество чисел не превышает 1000. Введённые числа не превышают 30 000.

Программа должна вывести одно число: количество чисел, кратных 4 и оканчивающихся на 2.

Пример работы программы:

Входные данные	Выходные данные
12 140 22 0	1

Номер: 57e4c1

Напишите программу, которая в последовательности натуральных чисел определяет сумму чисел, кратных 7. Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число, кратное 7.

Количество чисел не превышает 100. Введённые числа по модулю не превышают 300.

Программа должна вывести одно число: сумму чисел, кратных 7.

Пример работы программы:

Входные данные	Выходные данные
3 14 25 7	21

Номер: 577493

Напишите программу, которая в последовательности натуральных чисел определяет сумму чисел, оканчивающихся на 5. Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число, оканчивающееся на 5.

Количество чисел не превышает 100. Введённые числа по модулю не превышают 300.

Программа должна вывести одно число: сумму чисел, оканчивающихся на 5.

Пример работы программы:

Входные данные	Выходные данные
3 15 25 24	40

Номер: 31559e

Напишите программу, которая в последовательности натуральных чисел определяет количество чисел, кратных 8. Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число, кратное 8.

Количество чисел не превышает 1000. Введённые числа по модулю не превышают 30 000.

Программа должна вывести одно число: количество чисел, кратных 8.

Пример работы программы:

Входные данные	Выходные данные
3 16 26 24	2

Номер: 5D1c8B

Напишите программу, которая в последовательности натуральных чисел определяет максимальное число, оканчивающееся на 9. Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число, оканчивающееся на 9.

Количество чисел не превышает 1000. Введённые числа по модулю не превышают 30 000.

Программа должна вывести одно число: максимальное число, оканчивающееся на 9.

Пример работы программы:

Входные данные	Выходные данные
3 9 19 23	19

ПРИЛОЖЕНИЕ

Электронное приложение 1. Листинги программ, представленных в разделе «Сборник задач с решениями из открытого банка тестовых заданий с сайта ФГБНУ «Федеральный институт педагогических измерений»

Ссылка для скачивания материалов с Яндекс-диска:
<https://disk.360.yandex.ru/d/hZHnGMw-dj9prw>



Электронное приложение 2. Реализация практической части раздела «Язык программирования» учебников Л.Л. Босовой «Информатика» (8–9 классы)

Ссылка для скачивания материалов с Яндекс-диска:
<https://disk.360.yandex.ru/i/RZJHTd6rvIE2g>



Для заметок

Учебное издание

**МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ
по решению сложных задач
с использованием среды
программирования Python**

Подписано в печать xx.06.2025 г.
Гарнитура Times New Roman. Формат 60×84 1/16
Бумага офсетная. Усл. п. л. 4.42
Тираж 50 экз. Заказ № xxx/2025

Кировское областное государственное образовательное автономное
учреждение дополнительного профессионального образования
«Институт развития образования Кировской области»
610046, Кировская обл., г. Киров, ул. Романа Ердякова, д. 23, к. 2
Тел.: 8 (8332) 25-54-42 (доб. 301)
E-mail: rio@kirovipk.ru

Отпечатано в ООО «Полиграфовна»
610037, г. Киров, ул. Пархоменко, д. 9, помещение 1001
тел. 8 /8332/ 66-15-16, 66-15-15, 44-95-81