

**МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

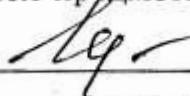
**Министерство образования Тульской области**

**Муниципальное образование Кимовский район**

**МКОУ СОШ № 4**

**РАССМОТРЕНО**

руководитель ШМО  
учителей-предметников

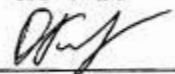


Шаханова Т.А.

Протокол № 1  
от «30». 08.2023 г.

**СОГЛАСОВАНО**

Заместитель директора  
по УВР



Келлер О. Н.

«30». 08.2023 г.

**УТВЕРЖДЕНО**

Директор школы



Яковлева С. А.

Приказ № 35  
от «01». 09.2023 г.

**ПРОГРАММА КУРСА ВНЕУРОЧНОЙ ДЕЯТЕЛЬНОСТИ  
«ИНФОРМАТИКА: СИЛЬНЫЙ СТАРТ»  
(10 класс)**

г. КИМОВСК, 2023

## ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

### **Актуальность и назначение программы**

Рабочая программа курса внеурочной деятельности «Углубленная информатика: сильный старт» разработана в соответствии с ФЗ «Об образовании в Российской Федерации», с Письмом Министерства образования и науки РФ от 28 октября 2015 г. № 08-1786 «О рабочих программах учебных предметов».

Актуальность курса обусловлена широким внедрением языка программирования Python для разработки различных приложений, позволяет работать с xml/html-файлами, http-запросами, изображениями, аудио- и видеофайлами, используется в робототехнике, программировании математических и научных вычислений, разработках искусственного интеллекта, что создает необходимость подготовки уверенных в завтрашнем дне и востребованных ИТ-специалистов. При изучении информатики на базовом уровне, на темы «алгоритмы» и «программирование» отводится недостаточное количество часов для формирования алгоритмического мышления. Обучение языку программирования Python создает благоприятные условия для «погружения» обучающихся в мир логики, математического моделирования, для интеллектуального и духовного воспитания личности ребенка, социально-культурного и профессионального самоопределения, развития познавательной активности и творческой самореализации.

Программа учитывает возрастные, психологические, физические особенности детей раннего юношеского возраста. Работа с обучаемыми строится на взаимосотрудничестве, на основе уважительного, искреннего, деликатного и тактичного отношения к личности ребенка. Важный аспект в обучении – индивидуальный подход, удовлетворяющий требованиям познавательной деятельности.

**Цель программы** – формирование цифровых навыков, в том числе ключевых компетенций цифровой экономики, таких как базовое программирование на Python, основы работы с данными, коммуникация в современных цифровых средах, информационная безопасность; воспитание ответственного и избирательного отношения к информации.

### **Задачи программы:**

1. Сформировать умения и навыки составления простых программ по построенному алгоритму на Python;
2. Сформировать умение грамотно интерпретировать результаты решения практических задач с помощью информационных технологий, применять полученные результаты в практической деятельности.

**Педагогическая целесообразность** программы состоит в том, что в процессе ее реализации обучающиеся научатся анализировать возникающие в командной работе ситуации; использовать полученные знания о специфических методах решения задач по информатике повышенной сложности, умение применять различные методики индивидуального и командного поиска

решения нестандартных задач; знать об особенностях применения различных языков программирования.

#### **Варианты реализации программы и формы проведения занятий**

Возраст обучающихся, участвующих в реализации программы: 15–17 лет (обучающиеся 10 класса).

Сроки реализации образовательной программы – 1 учебный год. Общий объем реализации программы – 34 часа.

Режим занятий – 1 раз в неделю.

Формы занятий – практикум, занятия-дискуссии.

Формы подведения итогов реализации программы – выполнение тестирования / выполнение проверочных работ / выполнение итогового тестирования.

#### **Взаимосвязь с программой воспитания**

Программа курса внеурочной деятельности разработана с учетом рекомендаций Примерной программы воспитания. Согласно Примерной программе воспитания у обучающегося должны быть сформированы ценности научного познания, способность к творческому созидательному социально значимому труду и готовность к получению профессионального образования. Реализация курса способствует осуществлению главной цели воспитания – полноценному личностному развитию школьников и созданию условий для их позитивной социализации.

#### **Особенности работы педагогов по программе**

В планировании, организации и проведении занятий может принимать участие как педагог из числа профессорско-преподавательского состава ТГПУ им. Л.Н. Толстого, так и педагог, ведущий уроки информатики в образовательной организации. Задача педагога состоит в реализации содержания курса через вовлечение обучающихся в многообразную деятельность, организованную в разных формах. Особенностью занятий является их интерактивность.

## **СОДЕРЖАНИЕ ПРОГРАММЫ**

### **Раздел 1. Знакомство с языком Python. Переменные и выражения**

Введение. Инструктаж по ТБ. Тестирование на определение базового уровня знаний. Современные языки программирования. Алгоритм. Язык программирования. Программа. Среда разработки IDE. Интерфейс Sculpt. Виды алгоритмов: линейный, разветвляющийся. Переменные. Правила образования имен переменных. Типы данных: целое число, строка. Функция. Виды функций. Функция: print(), input(), int(). Ветвление в Python. Оператор if-else. Вложенное ветвление. Множественное ветвление. Оператор if-elif-else. Проект «Чат-бот».

### **Раздел 2. Циклы**

Логическое выражение. Простые и сложные логические выражения. Результат вычисления логического выражения. Условие. Операции сравнения в Python. Логические операторы в Python: and, or и not. Операторы цело-

численного деления и деления с остатком на Python. Цикл с предусловием. Цикл с параметром. Проект «Максимум и минимум».

### **Раздел 3. Списки и словари**

Функции `str()` и `int()`. Методы для работы со строками. Создание списка в Python. Действия над элементами списка. Функции `append()`, `remove()`. Объединение списков. Циклический просмотр списка. Сортировка списков. Сумма элементов списка. Обработка списков. Сравнение списков и словарей. Словарь. Создание словаря в Python. Добавление новой записи в словарь. Вывод значения по ключу. Замена элемента словаря. Удаление элемента из словаря. Работа с элементами словаря. Методы работы со списками (`len()`, `clear()`, `keys()`, `values()`, `items()`).

### **Раздел 4. Знакомство с модулем Turtle. Функции и события на примере модуля Turtle**

Функции модуля Turtle. Самостоятельное создание функции. Глобальные и локальные переменные. Объект «экран». Событие. Работа с событиями. Фракталы. Рекурсия. Кривая Коха.

### **Раздел 5. Информационная безопасность**

Информационная безопасность. Приватность и защита персональных данных. Основные типы угроз в Интернете. Правила поведения в Интернете. Кибербуллинг. Защита частных данных. Финансовая информационная безопасность. Виды финансового мошенничества. Шифрование и криптография.

## **ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОСВОЕНИЯ КУРСА ВНЕУРОЧНОЙ ДЕЯТЕЛЬНОСТИ**

### **Предметные результаты:**

- развитие умений и навыков использования среды IDLE для разработки Python-программ;
- овладение синтаксисом основных конструкций языка программирования Python;
- овладение умениями и навыками работы с различными типами данных языка программирования Python;
- развитие навыков процедурного, а также объектно-ориентированного программирования.

### **Метапредметные:**

- развитие умений организации собственной учебной деятельности, включающих целеполагание, планирование, контроль;
- развитие умений постановки и формулирования проблемы, структурирования и визуализации информации, выбора наиболее эффективных способов решения.

### **Личностные результаты:**

- формирование целостного мировоззрения, соответствующего современному уровню развития науки и общественной практики;

- развитие познавательной активности, интеллектуальных и творческих способностей; развитие алгоритмического мышления;
- личностное и предпрофессиональное самоопределение.

## ТЕМАТИЧЕСКОЕ ПЛАНИРОВАНИЕ КУРСА

Раздел	Тема урока	Кол-во часов
Раздел 1. Знакомство с языком Python. Переменные и выражения	Современные языки программирования. Алгоритм. Язык программирования. Программа. Среда разработки IDE. Интерфейс Sculpt	1
	Виды алгоритмов: линейный, разветвляющийся. Переменные	1
	Правила образования имен переменных. Типы данных: целое число, строка	1
	Функция. Виды функций. Функция: print(), input(), int()	1
	Ветвление в Python. Оператор if-else. Вложенное ветвление. Множественное ветвление. Оператор if-elif-else	1
	Проект «Чат-бот»	1
Раздел 2. Циклы	Логическое выражение	1
	Простые и сложные логические выражения	1
	Простые и сложные логические выражения	1
	Результат вычисления логического выражения	1
	Условие. Операции сравнения в Python	1
	Условие. Операции сравнения в Python	1
	Логические операторы в Python: and, or и not	1
	Операторы целочисленного деления и деления с остатком на Python	1
	Цикл с предусловием	1
	Цикл с параметром	1
Проект «Максимум и минимум»	1	
Раздел 3. Списки и словари	Функции str() и int(). Методы для работы со строками. Создание списка в Python	1
	Действия над элементами списка. Функции append(), remove()	1
	Объединение списков. Циклический просмотр списка	1
	Сортировка списков. Сумма элементов	1

Раздел	Тема урока	Кол-во часов
	списка. Обработка списков	
	Сравнение списков и словарей	1
	Словарь. Создание словаря в Python	1
	Добавление новой записи в словарь. Вывод значения по ключу	1
	Замена элемента словаря. Удаление элемента из словаря. Работа с элементами словаря	1
	Методы работы со списками (len(), clear(), keys(), values(), items())	1
Раздел 4. Знакомство с модулем Turtle. Функции и события на примере модуля Turtle	Функции модуля Turtle	1
	Самостоятельное создание функции	1
	Глобальные и локальные переменные. Объект «экран»	1
	Событие. Работа с событиями. Фракталы. Рекурсия. Кривая Коха	1
Раздел 5. Информационная безопасность	Информационная безопасность. Приватность и защита персональных данных. Основные типы угроз в Интернете. Правила поведения в Интернете. Кибербуллинг	1
	Защита приватных данных. Финансовая информационная безопасность. Виды финансового мошенничества. Шифрование и криптография	1

## ПРИЛОЖЕНИЕ

**Методические материалы для ученика:** помодульные дидактические материалы, представленные на образовательной платформе (в том числе раздаточный материал и т. д.).

**Методические материалы для учителя:** методические материалы. Демонстрационные материалы по теме занятия. Методическое видео с подробным разбором материалов, рекомендуемых для использования на занятии.

**Цифровые образовательные ресурсы и ресурсы интернета:** образовательная платформа.

**Учебное оборудование:** компьютер (стационарный компьютер, ноутбук, планшет). Компьютерные мыши. Клавиатуры.

**Учебное оборудование для проведения лабораторных работ, практических работ и демонстраций:** мультимедийный проектор с экраном (интерактивной доской) или интерактивная панель.

**Литература и интернет-ресурсы для обучающихся**

1. Федоров, Д.Ю. Программирование на языке высокого уровня Python: учеб. пос. для сред. проф. обр. 3-е изд., перераб. и доп. М.: Изд-во Юрайт, 2021. 210 с. (Профессиональное образование). ISBN 978-5-534-12829-1 [Электронный ресурс]. Режим доступа: <https://urait.ru/bcode/487079>.
2. Чернышев, С.А. Основы программирования на Python: учеб. пос. для вузов. 2-е изд., перераб. и доп. М.: Изд-во Юрайт, 2021. 286 с. (Высшее образование). ISBN 978-5-534-14350-8. [Электронный ресурс]. Режим доступа: <https://urait.ru/bcode/477353>.
3. Кремлев, А.Г. Основные понятия теории игр: учеб. пос. Екатеринбург: Изд-во Урал. ун-та, 2016. 144 с.
4. Гниденко, И.Г., Павлов Ф.Ф., Федоров Д.Ю. Технологии и методы программирования: учебное пособие для вузов. М.: Изд-во Юрайт, 2023. 235 с. (Высшее образование). ISBN 978-5-534-02816-4 [Электронный ресурс]. Режим доступа: <https://urait.ru/viewer/tehnologii-i-metody-programmirovaniya-511891#page/2>.
5. Инфоурок: ведущий образовательный портал России [Электронный ресурс]. Режим доступа: <https://infourok.ru>.

## ПАКЕТ ОЦЕНОЧНЫХ МАТЕРИАЛОВ И КРИТЕРИИ ОЦЕНИВАНИЯ ПО ПРЕДМЕТУ

При подготовке к итоговой аттестации по курсу слушателям предлагаются групповые кейсы, состоящие из стандартных заданий курса «Информатика», решить которые предлагается нестандартным методом и объединить в единый пакет с организованным меню. Кроме групповых проектов осуществляется тестовый контроль.

### 1. ТЕСТОВЫЕ ЗАДАНИЯ

#### Тест по теме «Циклы»

1. Определите, что будет напечатано в результате выполнения следующего скрипта:

```
s = 0
for k in range(3,11):
    s = s + k
print(s)
```

2. Определите, что будет напечатано в результате выполнения следующего скрипта:

```
s = 0
for k in range(-5,11):
    s = s + 2 * k
print(s)
```

3. Определите, что будет напечатано в результате выполнения следующего скрипта:

```
s = 1
for k in range(1,30):
    s = (k - 5) * s
print(s)
```

4. Определите, что будет напечатано в результате выполнения следующего скрипта:

```
s = 1
for k in range(30):
    s = (-1) * s
print(s)
```

5. Определите, что будет напечатано в результате выполнения следующего скрипта:

```
z = 30
for n in range(30):
    if n > 10:
        z = z - n
    else:
        z = z + n
print(z)
```



6. Определите, что будет напечатано в результате выполнения следующего скрипта:

```
z = 30
for n in range(10):
    if n < 0:
        z = z - 2 * n
    else:
        z = n - z
print(z)
```

7. Определите, что будет напечатано в результате выполнения следующего скрипта:

```
a = 23
b = 4
while a > b:
    if a % 2 == 0:
        b = b + a
    else:
        a = a - 2 * b + 1
print(b)
```

8. Определите, что будет напечатано в результате выполнения следующего скрипта:

```
s = 0
m = 123
while m > 0:
    d = m % 10
    s = s + d
    m = m // 10
print(s)
```

9. Определите, что будет напечатано в результате выполнения следующего скрипта:

```
c = 0
m = 123
while m > 1:
    d = m % 10
    c = (c + d) * 10
    m = m // 10
print(c)
```

10. В результате выполнения программы, записанной ниже, на экран будет выведено два числа А и В. Укажите такое наибольшее число х, при вводе которого на экран будет выведено сначала 3, а потом 5:

```
x = int(input())
A = 0
B = 0
while x > 0:
    A = A + 1
```

```

if B < x % 10:
    B = x % 10
x = x // 10
print(A)
print(B)

```

### Ответы к тесту по теме «Циклы»

Вопрос	1	2	3	4	5	6	7	8	9	10
Ответ	52	80	0	1	-295	-35	20	6	320	555

## 2. ПРОВЕРОЧНАЯ РАБОТА «PYTHON. ВВОД-ВЫВОД ДАННЫХ»

### I часть – теоретическая

- Какая команда используется для вывода (печати) данных?
  - Console.WriteLine()
  - cout()
  - printf()
  - print()
- Выберите верные строки кода.
  - print('Просто текст...')
  - print()
  - print("I'm a math teacher and a programmer!")
  - print('Поэма "Мертвые души" одна из самых интересных')
  - print("3.1415")
  - print('I'm 16 and I'm from Northern Ireland.')
- Выберите верные строки кода.
  - print("10", '100', '1000')
  - print('Python', 'is the best', '!!')
  - print("раз", "два", "три")
  - print("Python", , "is the best")
- Что выведет следующий код print('1', '2', '3', '4', sep='\*')?
  - 1 2 3 4
  - 1234
  - 1\*2\*3\*4
  - 24
- Выберите верные строки кода.
  - print("Honey, what's your hurry", end='?')
  - print("The world's a little blurry", "Or maybe it's my eyes", end='!!!', sep=' :')
   
)

- c. `print("Told you not to worry", "But maybe that's a lie", sep=':')`
  - d. `print("Remember not to get too close to stars", "They're never gonna give you love like ours", sep=" ")`
  - e. `print("Remember not to get too close to stars", end="")`
6. Какая команда используется для считывания данных с клавиатуры?
- a. `Console.ReadLine()`
  - b. `scanf()`
  - c. `cin`
  - d. `input()`
7. Какая из указанных строк считывает целое число в переменную n?
- a. `n = input()`
  - b. `n = integer(input())`
  - c. `n = int(input())`
  - d. `n = number(input())`
  - e. `n = str(input())`
8. Выберите верные утверждения.
- a. Имя переменной может начинаться с символа подчеркивания (`_`)
  - b. Имя переменной не может оканчиваться цифрой
  - c. Имя переменной не может совпадать с ключевым (зарезервированным) словом
  - d. Имя переменной не может начинаться с цифры
9. Какое число выведет следующий код?
- ```
s = 13
k = -5
d = s + 2
s = d
k = 2 * s
print(s + k + d)
```
10. Какое число выведет следующий код?
- ```
a = 17 // (23 % 7)
b = 34 % a * 5 - 29 % 4 * 3
print(a * b)
```

## II часть – практическая

### Задача 1

Напишите программу, которая выводит прямоугольник, по периметру состоящий из звездочек (\*).

**Примечание.** Высота и ширина прямоугольника равны 4 и 17 звездочкам соответственно.

## Тестовые данные

---

### Sample Input:

### Sample Output:

```
*****  
* *  
* *  
*****
```

## Задача 2

Напишите программу, которая считывает два целых числа  $a$  и  $b$  и выводит на экран квадрат суммы  $(a+b)^2$  и сумму квадратов  $a^2+b^2$  этих чисел.

### Формат входных данных

На вход программе подается два целых числа, каждое – на отдельной строке.

### Формат выходных данных

Программа должна вывести текст в соответствии с условием.

## Тестовые данные

---

### Sample Input 1:

```
3  
2
```

### Sample Output 1:

```
Квадрат суммы 3 и 2 равен 25  
Сумма квадратов 3 и 2 равна 13
```

### Sample Input 2:

```
-5  
1
```

### Sample Output 2:

```
Квадрат суммы -5 и 1 равен 16  
Сумма квадратов -5 и 1 равна 26
```

### Sample Input 3:

```
17  
0
```

### Sample Output 3:

```
Квадрат суммы 17 и 0 равен 289  
Сумма квадратов 17 и 0 равна 289
```

## Задача 3

Как известно, целые числа в языке Python не имеют ограничений, которые встречаются в других языках программирования. Напишите программу, которая считывает четыре целых положительных числа  $a$ ,  $b$ ,  $c$  и  $d$  и выводит на экран значение выражения  $a^b + c^d$ .

### Формат входных данных

На вход программе подается четыре целых положительных числа  $a$ ,  $b$ ,  $c$  и  $d$ , каждое на отдельной строке в указанном порядке.

### Формат выходных данных

Программа должна вывести значение  $a^b + c^d$ .

Тестовые данные

---

#### Sample Input:

9  
29  
7  
27

---

#### Sample Output:

4710194409608608369201743232

---

### Задача 4

Напишите программу, которая считывает целое положительное число  $n$ ,  $n \in [1;9]$  и выводит значение числа  $n + \overline{nn} + \overline{nnn}$ .

### Формат входных данных

На вход программе подается одно целое положительное число  $n$ ,  $n \in [1;9]$ .

### Формат выходных данных

Программа должна вывести число  $n + \overline{nn} + \overline{nnn}$ .

**Примечание.** Для первого теста  $1 + 11 + 111 = 123$ .

Тестовые данные

---

#### Sample Input 1:

1

---

#### Sample Output 1:

123

---

#### Sample Input 2:

2

---

#### Sample Output 2:

246

---

#### Sample Input 3:

3

---

#### Sample Output 3:

369

---

## Ответы

### 1 часть

1	d
2	bcde
3	b
4	c
5	abc
6	d
7	c
8	acd
9	60
10	56

### 2 часть

Один из вариантов написания программ (решение может быть и другим):

#### Задача 1

```
print('*****')
print('* *')
print('* *')
print('*****')
```

#### Задача 2

```
a = int(input())
b = int(input())
print('Квадрат суммы', a, 'и', b, 'равен', (a + b) ** 2)
print('Сумма квадратов', a, 'и', b, 'равна', a ** 2 + b ** 2)
```

#### Задача 3

```
a=int(input())
b=int(input())
c=int(input())
d=int(input())
print(a**b+c**d)
```

#### Задача 4

```
n=int(input())
nn=n*10+n
nnn=n*100+n*10+n
print(n+nn+nnn)
```

## 3. ПРИМЕРЫ ЗАДАЧ ДЛЯ ПРОЕКТОВ УЧАЩИХСЯ С РЕШЕНИЯМИ

#### Задача 1

(№ 3770) (П. Волгин) Рассматривается множество целых чисел, принадлежащих числовому отрезку  $[100; 1000000]$ , которые удовлетворяют следующим условиям:  
а) Число в шестнадцатеричной записи оканчивается цифрой «А»;  
б) Число в шестнадцатеричной записи начинается цифрой «В»;  
в) Число не делится на 12.  
Найдите количество таких чисел и максимальное из них. В ответе запишите сначала количество, а потом максимальное число.

**Решение:**

```
count = 0
for i in range(100, 1000000 + 1):
    if i % 16 == 10:
        t = i
        while t != 0:
            a = t % 16
            t //= 16
            if a == 11:
                if i % 12 != 0:
                    count += 1
                    m = i
print(count, m)
```

**Ответ:**

4369 786426

#### Задача 2

(№ 3845) (П. Волгин) Рассматривается множество целых чисел, принадлежащих числовому отрезку  $[255; 4095]$ , которые удовлетворяют следующим условиям:  
а) Число в троичной записи содержит одну цифру «1» или две цифры «0»;  
б) Число делится на 2 и 5, но не делится на 20.  
Найдите количество таких чисел и их сумму. В ответе запишите сначала количество, а затем сумму.

**Решение:**

```
count = s = 0
for i in range(255, 4095 + 1):
    s3 = ''
    t = i
    while t != 0:
        s3 += str(t % 3) + s3
        t //= 3
    if s3.count('1') == 1 or s3.count('0') == 2:
        if i % 2 == 0 and i % 5 == 0:
            if i % 20 != 0:
                count += 1
                s += i
print(count, s)
```

**Ответ:**

60 133320

#### Задача 4

(№ 3990) (П. Волгин) Рассматривается множество целых чисел, принадлежащих числовому отрезку  $[10101; 11110]$ , которые удовлетворяют следующим условиям:  
а) Число делится на  $100_2, 100_8, 100_{16}$ .  
б) Число не делится на  $110_2, 12_3, 3A_{16}$ .  
Найдите количество таких чисел и минимальное из них. В ответе запишите сначала количество, а затем минимальное число.

**Решение:**

```
count = 0
mi = 11110 + 1
for i in range(10101, 11110 + 1):
    if i % int('100', 2) == 0:
        if i % int('100', 8) == 0:
            if i % int('100', 16) == 0:
                if i % int('110', 2) != 0:
                    if i % int('12', 3) != 0:
                        if i % 58 != 0:
                            count += 1
                            mi = min(mi, i)
print(count, mi)
```

**Ответ:**

2 10496

### Задача 1

(№ 2343) (А.Г. Минин) Рассматривается множество целых чисел, принадлежащих числовому отрезку [2079; 43167], которые делятся на 7 и обязательно имеют в своей записи цифры 0, 2 и 5. Найдите количество таких чисел и минимальное из них. В ответе запишите два целых числа: сначала количество, затем минимальное число.

#### Решение:

```
count = 0
mi = 43167 + 1
for i in range(2079, 43167 + 1):
    if i % 7 == 0:
        t = str(i)
        if '0' in t and '2' in t and '5' in t:
            count += 1
            mi = min(mi, i)
print(count, mi)
```

#### Ответ:

233 2205

### Задача 3

(№ 2742) (Е. Дюбе) Рассматривается множество целых чисел, принадлежащих числовому отрезку [333666; 666999], которые удовлетворяют следующим условиям: а) в числе есть не менее двух цифр 7; б) число кратно 17. Найдите наибольшее из таких чисел и их количество.

#### Решение:

```
count = 0
for i in range(333666, 666999 + 1):
    t = i
    count7 = 0
    while t != 0:
        if t % 10 == 7:
            count7 += 1
            if count7 >= 2:
                break
        t //= 10
    if count7 >= 2:
        if i % 17 == 0:
            count += 1
            m = i
print(m, count)
```

#### Ответ:

666774 1538



### Задача 1

«А»: Напишите программу, которая получает два целых числа  $A$  и  $B$  ( $0 < A < B$ ) и выводит квадраты всех натуральных чисел в интервале от  $A$  до  $B$ .

Пример:

```
Введите два целых числа:  
10 12  
10*10=100  
11*11=121  
12*12=144
```

Решение:

```
a, b = map(int, input().split())  
while a <= b:  
    print(a, '**', a, '=', a ** 2)  
    a += 1
```

### Задача 2

Логическая функция  $F$  задается выражением:

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

Дан частично заполненный фрагмент, содержащий неповторяющиеся строки таблицы истинности функции  $F$ .

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

Переменная 1	Переменная 2	Переменная 3	Переменная 4	Функция
???	???	???	???	$F$
1	1		1	1
	1	1		1
1	1			1

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

Решение:

```
print('x y z w')  
for x in range(2):  
    for y in range(2):  
        for z in range(2):  
            for w in range(2):  
                if (not y or z) and not(not(y or w) or (z and  
x)):  
                    print(x, y, z, w)
```

Ответ: zwxу

### Задача 8

Валерия составляет 3-буквенные коды из букв В, А, Л, Е, Р, И, Я, причём буква В должна входить в код ровно один раз. Все полученные коды Валерия записала в алфавитном порядке и пронумеровала. Начало списка выглядит так:

1. AAB
2. ABA
3. ABE

...

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

Решение:

```
alf = ['B', 'A', 'L', 'E', 'P', 'I', 'Y']
words = []
for a in alf:
    for b in alf:
        for c in alf:
            s = a + b + c
            if s.count('B') == 1:
                words.append(s)
words.sort()
print(words)
for i in range(len(words)):
    if words[i].count('A') == 0:
        print(i + 1)
        break
```

Ответ: 20

### Задача 14

Значение выражения  $729^6 + 3^{14} - 36$  записали в системе счисления с основанием 9. Сколько раз в этой записи встречается цифра 0?

Решение:

```
a = 729 ** 6 + 3 ** 14 - 36
s = ''
while a != 0:
    s += str(a % 9)
    a //= 9
print(s.count('0'))
```

Ответ: 12

### Задача 15

Обозначим через  $\text{ДЕЛ}(n, m)$  утверждение «натуральное число  $n$  делится без остатка на натуральное число  $m$ ».

Для какого наименьшего натурального числа  $A$  формула

$\text{ДЕЛ}(A, 40) \wedge (\text{ДЕЛ}(780, x) \rightarrow (\neg \text{ДЕЛ}(A, x) \rightarrow \neg \text{ДЕЛ}(180, x)))$

тождественно истинна, то есть принимает значение 1 при любом натуральном  $x$ ?

Решение:

```
A = 1
while True:
    count = 0
    for x in range(1, 1000 + 1):
        if (A % 40 == 0) and (not 780 % x == 0 or A % x == 0
or not 180 % x == 0):
            count += 1
    if count == 1000:
        print(A)
        break
    A += 1
```

Ответ: 120

### Задача 16

Обозначим через  $\text{mod}(a, b)$  остаток от деления натурального числа  $a$  на натуральное число  $b$ . Алгоритм вычисления значения функции  $F(n)$ , где  $n$  – целое неотрицательное число, задан следующими соотношениями:

$$F(0) = 0;$$

$$F(n) = F(n/3), \text{ если } n > 0 \text{ и при этом } \text{mod}(n, 3) = 0;$$

$$F(n) = \text{mod}(n, 3) + F(n - \text{mod}(n, 3)), \text{ если } \text{mod}(n, 3) > 0.$$

Назовите минимальное значение  $n$ , для которого  $F(n) = 9$ .

**Решение:**

```
def f(n):
    if n == 0:
        return 0
    else:
        if n % 3 == 0:
            return f(n / 3)
        else:
            return n % 3 + f(n - n % 3)

for i in range(1, 1000):
    if f(i) == 9:
        print(i)
        break
```

**Ответ: 161**

### Задача 17

Назовём натуральное число подходящим, если у него ровно 3 различных простых делителя. Например, число 180 подходящее (его простые делители – 2, 3 и 5), а число 12 – нет (у него только два различных простых делителя). Определите количество подходящих чисел, принадлежащих отрезку [50 001; 90 000], а также наименьшее из таких чисел. В ответе запишите два целых числа: сначала количество, затем наименьшее число.

**Решение:**

```
def allDivs(x):
    divs = []
    for d in range(2, int(x ** 0.5) + 1):
        if x % d == 0:
            divs += [d]
            if x // d > d:
                divs += [x // d]
    return sorted(divs)

def p(x):
    for d in range(2, int(x ** 0.5) + 1):
        if x % d == 0:
            return False
    return True

count = 0
m = 90000
for i in range(50001, 90000 + 1):
    k = 0
    divs = allDivs(i)
    for j in divs:
        if p(j):
            k += 1
            if k > 3:
                break
    if k == 3:
        count += 1
        if i < m:
            m = i
print(count, m)
```

**Ответ: 15587 50001**

### Задача 19

Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежат две кучи камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может добавить в одну из куч **один камень** или увеличить количество камней в куче в **два раза**. Например, пусть в одной куче 5 камней, а в другой 9 камней; такую позицию мы будем обозначать (5, 9). За один ход из позиции (5, 9) можно получить любую из четырёх позиций: (6, 9), (10, 9), (5, 10), (5, 18). Чтобы делать ходы, у каждого игрока есть неограниченное количество камней. Игра завершается в тот момент, когда суммарное количество камней в кучах становится не менее 93. Победителем считается игрок, сделавший последний ход, то есть первым получивший позицию, в которой в кучах будет 93 или больше камней. В начальный момент в первой куче было 12 камней, во второй куче –  $S$  камней,  $1 \leq S \leq 80$ .

**Решение:**

```
def f(x, y, p):
    if x + y >= 93 and p == 3:
        return True
    else:
        if x + y < 93 and p == 3:
            return False
    return f(x + 1, y, p + 1) or f(x * 2, y, p + 1) or \
        f(x, y + 1, p + 1) or f(x, y * 2, p + 1)

for i in range(1, 80 + 1):
    if f(12, i, 1):
        print(i, ' ')
        break
```

**Ответ: 21**

### Задача 20

Для игры, описанной в задании 19, найдите все такие значения  $S$ , при которых у Пети есть выигрышная стратегия, причём Петя не может выиграть первым ходом, но может выиграть своим вторым ходом независимо от того, как будет ходить Ваня. Найденные значения запишите в ответе в порядке возрастания.

**Решение:**

```
def f(x, y, p):
    if x + y >= 93 and p == 4:
        return True
    else:
        if x + y < 93 and p == 4:
            return False
        else:
            if x + y >= 93:
                return False
    if p % 2 == 1:
        return f(x + 1, y, p + 1) or f(x * 2, y, p + 1) or \
            f(x, y + 1, p + 1) or f(x, y * 2, p + 1)
    else:
        return f(x + 1, y, p + 1) and f(x * 2, y, p + 1) and \
            f(x, y + 1, p + 1) and f(x, y * 2, p + 1)

for i in range(1, 80 + 1):
    if f(12, i, 1):
        print(i, ' ')
```

**Ответ: 20 34 39**

### Задача 23

Исполнитель преобразует число на экране.  
У исполнителя есть три команды, которым присвоены номера:  
1. Прибавить 1  
2. Умножить на 2  
3. Умножить на 3  
Первая команда увеличивает число на экране на 1, вторая умножает его на 2, третья – умножает на 3.  
Программа для исполнителя – это последовательность команд.  
Сколько существует программ, которые преобразуют исходное число 2 в число 39, и при этом траектория вычислений содержит число 13 и не содержит числа 30?

Решение:

```
def f(x, y):  
    if x == y:  
        return 1  
    if x > y or x == 30:  
        return 0  
    if x < y:  
        return f(x + 1, y) + f(x * 2, y) + f(x * 3, y)  
  
print(f(2, 13) * f(13, 39))
```

Ответ: 75

### Задача 24

Текстовый файл содержит строки различной длины. Общий объем файла не превышает 1 Мбайт. Строки содержат только заглавные буквы латинского алфавита (ABC...Z).  
Необходимо найти строку, содержащую наименьшее количество букв N (если таких строк несколько, надо взять ту, которая находится в файле раньше), и определить, какая буква встречается в этой строке чаще всего. Если таких букв несколько, надо взять ту, которая позже стоит в алфавите.

Решение:

```
f = open('24.txt')  
m = 10000  
for s in f:  
    if s.count('N') < m:  
        m = s.count('N')  
f.close()  
  
f = open('24.txt')  
for s in f:  
    if s.count('N') == m:  
        a = [0] * 100  
        for i in range(len(s)):  
            a[ord(s[i])] += 1  
        break  
for i in reversed(range(len(a))):  
    if a[i] == max(a):  
        print(chr(i))  
        break
```

Ответ: Y

### Задача 25

Найдите все натуральные числа, принадлежащие отрезку  $[45\,000\,000; 50\,000\,000]$ , у которых ровно пять различных нечётных делителей (количество чётных делителей может быть любым). В ответе перечислите найденные числа в порядке возрастания.

Решение:

```
def p(x):
    for d in range(2, int(x ** 0.5) + 1):
        if x % d == 0:
            return False
    return True

for i in range(3, 1000):
    if p(i):
        d = 1
        while (i ** 4) * d <= 50000000:
            if (i ** 4) * d >= 45000000:
                print((i ** 4) * d)
            d *= 2
```

Ответ:

45212176  
45265984  
47458321  
48469444

### Задача 26

В текстовом файле записан набор натуральных чисел, не превышающих  $10^9$ . Гарантируется, что все числа различны. Необходимо определить, сколько в наборе таких пар нечётных чисел, что их среднее арифметическое тоже присутствует в файле, и чему равно наибольшее из средних арифметических таких пар.

Решение:

```
f = open('26.txt', 'r')
n = int(f.readline())
a = [int(f.readline()) for i in range(n)]
a.sort()
count = m = 0
for i in range(n - 1):
    for j in range(i + 1, n):
        if a[i] % 2 != 0:
            if a[j] % 2 != 0:
                s = (a[i] + a[j]) // 2
                L = i
                R = j
                while L < R - 1:
                    c = (L + R) // 2
                    if s < a[c]:
                        R = c
                    else:
                        L = c
                if a[L] == s:
                    count += 1
                    m = max(m, s)
    print (count, m)
```

Ответ: 14 679730035

### Задача 27

В текстовом файле записан набор натуральных чисел, не превышающих  $10^6$ . Гарантируется, что все числа различны. Из набора нужно выбрать три числа, сумма которых делится на 3. Какую наименьшую сумму можно при этом получить?

Решение:

```
f = open('27-B.txt')
n = int(f.readline())
a0 = []
a1 = []
a2 = []
for i in range(n):
    x = int(f.readline())
    if x % 3 == 0:
        a0 += [x]
    if x % 3 == 1:
        a1 += [x]
    if x % 3 == 2:
        a2 += [x]
a0.sort()
a1.sort()
a2.sort()
print(min(a0[0] + a0[1] + a0[2], a1[0] + a1[1] + a1[2], a0[0]
+ a1[0] + a2[0], a2[0] + a2[1] + a2[2]))
```

Ответ: 327 19851

Задача 2

«В»: Напишите программу, которая получает два целых числа и находит их произведение, не используя операцию умножения. Учтите, что числа могут быть отрицательными.

Пример:

```
Введите два числа:
10 -15
10 * (-15) = -150
```

Решение:

```
a, b = map(int, input().split())
k = 0
if a < 0 and b < 0:
    a1 = abs(a)
    b1 = abs(b)
else:
    if a * b < 0:
        k = 1
        a1 = abs(a)
        b1 = abs(b)
x = a1
n = 1
while n < b1:
    x += a1
    n += 1
if k == 1:
    x = -x
print(a, '*', b, '=', x)
else:
    m2 = d
if m1 > m2:
    m2 = m1
if e > m2:
    m2 = e
print(m2)
```