

Введение в программирование на языке Python

Проект «Отличный код»*



Содержание

3	Условия. Условный оператор	2
3.1	Условный оператор с одной ветвью	4
3.2	Условный оператор с двумя ветвями	5
3.3	Вложенные условные операторы	6
3.4	Каскадные условные инструкции	7
3.5	Примеры решения задач	8
3.5.1	Поиск максимума, минимума	8
3.5.2	Количество, сумма, произведение. Цифры целого числа	9
3.6	Тернарный оператор	12
	Компьютерный практикум	13
	Подсказки к решениям	15
	Ответы и решения	17
	Готовимся к ЕГЭ вместе	20

* Тестовый вариант (возможны опечатки). Нулевой курс мехмата ЮФУ. Проект «Отличный код», 2023–2026. Автор: кандидат физико-математических наук, Елена Владимировна Ширяева, <https://t.me/EVShiryaeva>

3 Условия. Условный оператор

Условие — это выражение, про которое можно сказать истинно оно или ложно (такие выражения называются логическими). Истинность выражения обозначается словом True. Ложность выражения обозначается словом False.

Примеры простых условий:

`myClass > 10` — значение переменной `myClass` больше 10 (True, если верно)

`city == 'Ростов'` — значение переменной `city` равно Ростов (True, если верно)

Простые условия строятся с помощью операций сравнения, см. таблицу 1.

Операции сравнения	Описание
<code>==</code>	равно
<code>!=</code>	не равно
<code>></code> , <code><</code>	больше, меньше
<code>>=</code>	больше или равно
<code><=</code>	меньше или равно

Таблица 1. Операции сравнения



Рис. 1. `cat = sausage`
Кот присваивает сосиску

Не путайте:
`=` (присвоить);
`==` (сравнить)



Рис. 2. `duck1 == duck2`
Сравнение уток

Выражение	Запись на языке Python
$x > 0$	<code>x > 0</code>
$x = 0$	<code>x == 0</code>
$x \neq 0$	<code>x != 0</code>
$x \geq 0$	<code>x >= 0</code>
$x \in [a; b]$	<code>a <= x <= b</code> или <code>x in range(a, b+1)</code>
x двузначное	<code>10 <= abs(x) <= 99</code>
n чётное	<code>n % 2 == 0</code>
n нечётное	<code>n % 2 != 0</code>
n кратно k	<code>n % k == 0</code>
n не кратно k	<code>n % k != 0</code>
n является полным квадратом	<code>int(n**0.5)**2 == n</code>

Таблица 2. Примеры логических выражений (условий) и их Python-запись

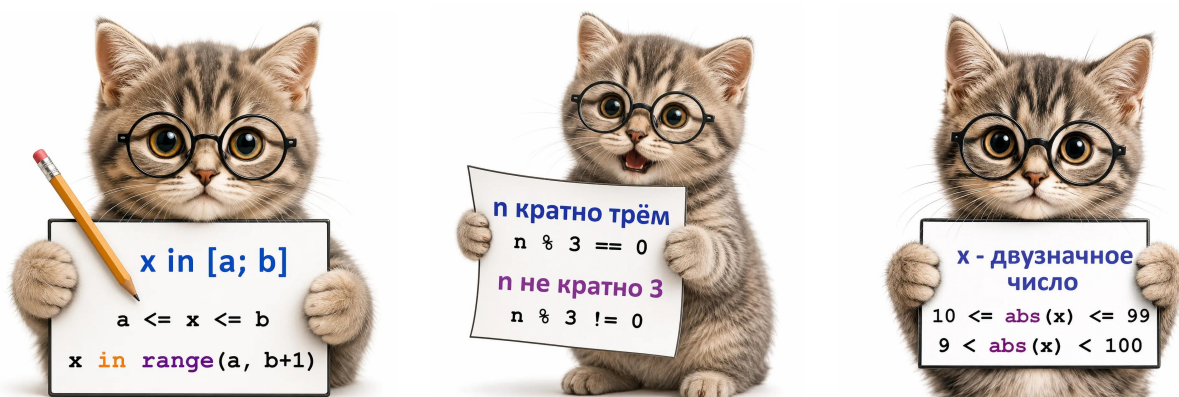


Рис. 3. Запись условий в Python

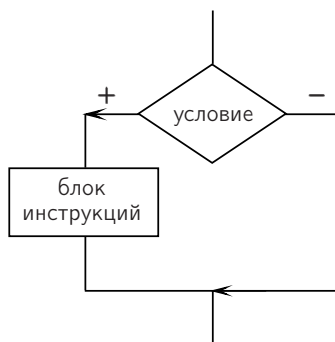
Выражение	Дополняющее выражение
True	False
False	True
$x > 0$	$x \leq 0$
$x = 0$	$x \neq 0$
$x \neq 0$	$x = 0$
$x \geq 0$	$x < 0$
$x \in [a; b]$	$x \notin [a; b]$
n чётное	нечётное
n не кратно k	кратно k

Таблица 3. Примеры логических выражений и их дополнений



Рис. 4. Чёрное дополняет белое в монохромном мире

3.1 Условный оператор с одной ветвью



Синтаксис условного оператора с одной ветвью

```
if условие:  
    блок инструкций
```

Если условие истинно, то выполняется блок инструкций (группа команд); если же условие ложно, то блок инструкций не будет выполнен и управление передаётся следующему оператору программы.

В языке Python для выделения блоков инструкций используются отступы. Все инструкции, которые относятся к одному блоку, должны иметь равную величину отступа слева (4 пробела).

Синтаксис условного оператора и отступы

```
if условие:  
    блок инструкций  
операторы, не принадлежащие блоку инструкций if  
# располагаются под словом if
```

Простые примеры использования условного оператора.

Программа

```
class_ = int(input('Введите свой класс >> '))  
if class_ < 10:  
    print('Вам надо на другие курсы')
```

Результаты двух запусков

```
Введите свой класс >> 9  
Вам надо на другие курсы  
===== RESTART =====  
Введите свой класс >> 11  
# Программа ничего не выведет!
```

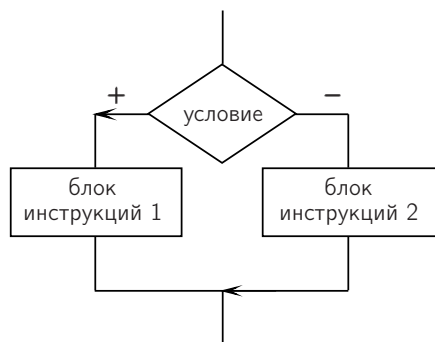
Программа

```
n = int(input('Введите отриц. число >> '))  
if n < 0:  
    print('Попали в тело усл. оператора')  
print('Оператор ЗА условным оператором')
```

Результаты двух запусков

```
Введите отриц. число >> -7  
Попали в тело усл. оператора  
Оператор ЗА условным оператором  
===== RESTART =====  
Введите отриц. число >> 0  
Оператор ЗА условным оператором
```

3.2 Условный оператор с двумя ветвями



Синтаксис условного оператора с двумя ветвями

```
if условие:
    блок инструкций 1
else:
    блок инструкций 2
```

Если условие истинно, то выполняется блок инструкций 1, иначе (т.е. истинно дополняющее условие) выполняется блок инструкций 2.

Пример программы

```
n = int(input('Введите число >> '))
if n < 0:
    print('Отрицательное число')
else:
    print('Неотрицательное число')
```

Результаты двух запусков

```
Введите число >> -5
Отрицательное число
===== RESTART =====
Введите число >> 0
Неотрицательное число
```

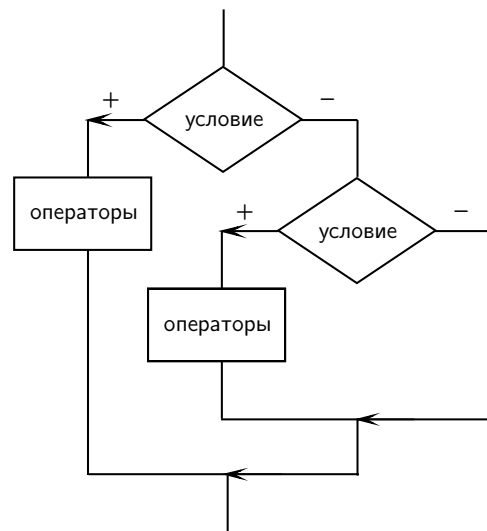
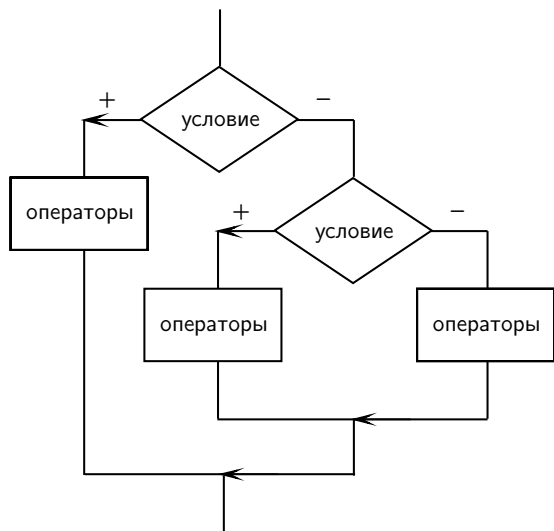
Пример программы

```
n = int(input('Введите число >> '))
if int(n**0.5)**2 == n:
    print(f'Полный квадрат числа {int(n**0.5)}')
else:
    print('НЕ является полным квадратом')
```

Результаты двух запусков

```
Введите число >> 100
Полный квадрат числа 10
===== RESTART =====
Введите число >> 50
НЕ является полным квадратом
```

3.3 Вложенные условные операторы



if-else в if-else

```

if условие 1 выполнено:
    блок инструкций 1
else:
    if условие 2 выполнено:
        блок инструкций 2
    else:
        блок инструкций 3
  
```

if в if-else

```

if условие 1 выполнено:
    блок инструкций 1
else:
    if условие 2 выполнено:
        блок инструкций 2
  
```

Пример программы (if-else в if-else)

```

n = int(input('Введите число >> '))
if n < 0:
    print('Отрицательное число')
else:
    if n > 0:
        print('Положительное число')
    else:
        print('Ноль')
  
```

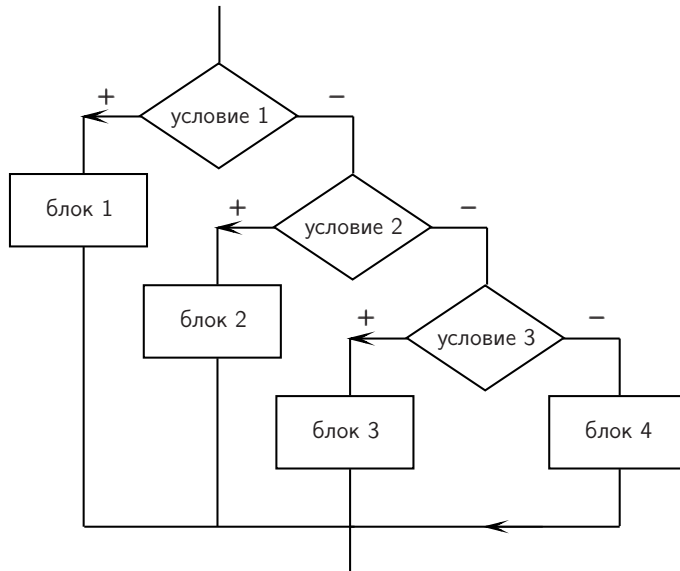
Результаты трёх запусков

```

Введите число >> -8
Отрицательное число
===== RESTART =====
Введите число >> 10
Положительное число
===== RESTART =====
Введите число >> 0
Ноль
  
```

3.4 Каскадные условные инструкции

В каскадной условной инструкции условия if, ..., elif проверяются по очереди. При истинности какого-то из условий выполняется соответствующий блок. Если все проверяемые условия ложны, то выполняется блок else, если он присутствует.



if-else в if-else

```
if условие 1 истинно:  
    блок 1  
elif условие 2 истинно:  
    блок 2  
elif условие 3 истинно:  
    блок 3  
else:  
    блок
```



Каскадные условные инструкции позволяют нагляднее представить структуру проверки условий, если их больше двух.

Пример программы

```
class_ = int(input('Класс >> '))  
if class_ == 11:  
    print('Срочно начинай готовиться к ЕГЭ')  
elif class_ == 10:  
    print('Начинай готовиться к ЕГЭ')  
elif class_ == 9:  
    print('Начинай готовиться к ОГЭ')  
else:  
    print('Наслаждайся пока детством')
```

Результаты четырёх запусков

```
Класс >> 11  
Срочно начинай готовиться к ЕГЭ  
===== RESTART =====  
Класс >> 10  
Начинай готовиться к ЕГЭ  
===== RESTART =====  
Класс >> 9  
Начинай готовиться к ОГЭ  
===== RESTART =====  
Класс >> 5  
Наслаждайся пока детством
```

3.5 Примеры решения задач

3.5.1 Поиск максимума, минимума

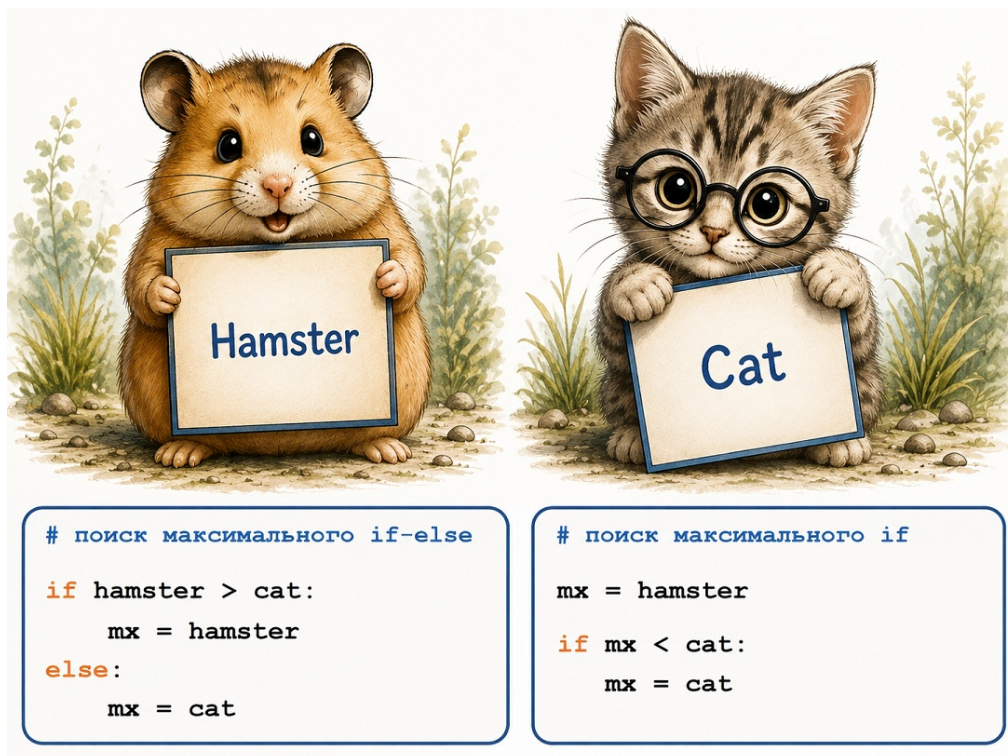


Рис. 5. Поиск максимума

Пример 3.1 (поиск наибольшего/наименьшего из двух чисел). Ввести два целых числа x и y . Найти наибольшее/наименьшее из них.

Поиск $\max(x, y)$. (if)

```
x = int(input('x = '))
y = int(input('y = '))
mx = x
if y > mx:
    mx = y
print('max =', mx)
```

Поиск $\max(x, y)$. (if-else)

```
x = int(input('x = '))
y = int(input('y = '))
if x > y:
    mx = x
else:
    mx = y
print('max =', mx)
```

Поиск $\min(x, y)$. (if)

```
x = int(input('x = '))
y = int(input('y = '))
mn = x
if y < mn:
    mn = y
print('min =', mn)
```

Поиск $\min(x, y)$. (if-else)

```
x = int(input('x = '))
y = int(input('y = '))
if x < y:
    mn = x
else:
    mn = y
print('min =', mn)
```

Пример 3.2 (одновременный поиск наибольшего и наименьшего из двух ~~котиков~~ чисел). Ввести два различных целых числа x и y . Присвоить переменной mx большее из этих чисел, а mn — меньшее.

Поиск $\max(x, y)$ и $\min(x, y)$. (if)

```
x = int(input('x = '))
y = int(input('y = '))
mx, mn = x, y
if x < y:
    mx, mn = y, x
print(f'{mx = }')
print(f'{mn = }')
```



Поиск $\max(x, y)$. (if-else)

```
x = int(input('x = '))
y = int(input('y = '))
if x > y:
    mx, mn = x, y
else:
    mx, mn = y, x
print(f'{mx = }')
print(f'{mn = }')
```

3.5.2 Количество, сумма, произведение. Цифры целого числа

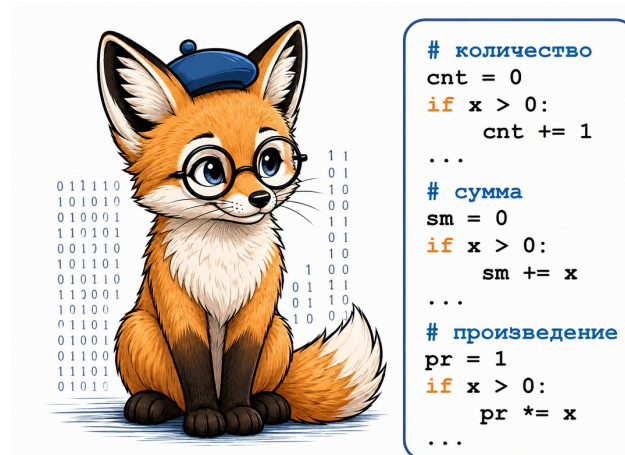


Рис. 6. Количество (сумма, произведение) положительных чисел

Пример 3.3. Среди заданных натуральных чисел x, y, z найти количество чётных чисел.

Количество чётных чисел

```
x = int(input('x = '))
y = int(input('y = '))
z = int(input('z = '))

cnt = 0 # инициализация счётчика чётных чисел
if x % 2 == 0:
    cnt += 1
if y % 2 == 0:
    cnt += 1
if z % 2 == 0:
    cnt += 1
print(cnt)
```

Пример 3.4. С клавиатуры вводятся целые числа x , y , z . Найти произведение ненулевых чисел.

Произведение ненулевых чисел

```
x = int(input('x = '))
y = int(input('y = '))
z = int(input('z = '))

pr = 1 # инициализация произведения чисел
if x != 0:
    pr *= x
if y != 0:
    pr *= y
if z != 0:
    pr *= z
print(pr)
```

Пример 3.5. С клавиатуры вводится трёхзначное натуральное число. Найти сумму нечётных цифр числа.

Сумма нечётных цифр числа

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

# отделение цифр натурального числа
n0 = n % 10
n1 = n // 10 % 10
n2 = n // 100

sm = 0 # инициализация суммы цифр
if n0 % 2 != 0:
    sm += n0
if n1 % 2 != 0:
    sm += n1
if n2 % 2 != 0:
    sm += n2

print(sm)
```

Результаты трёх запусков

```
n = 789
16
===== RESTART =====
n = 462
0
===== RESTART =====
n = 135
9
```



Пример 3.6. С клавиатуры вводится натуральное число. Если оно трёхзначное, то найти сумму цифр числа, если двузначное — найти произведение ненулевых цифр числа. Во всех других случаях вывести текст «Конец программы».

Программа

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

if 100 <= n <= 999: # число трёхзначное
    n0 = n % 10
    n1 = n // 10 % 10
    n2 = n // 100
    sm = n0 + n1 + n2
    print(f'{sm = }')
elif 10 <= n <= 99: # число двузначное
    n0 = n % 10
    n1 = n // 10 % 10
    if n0 == 0:
        pr = n1 # см. примечание
    else:
        pr = n0*n1
    print(f'{pr = }')
else:
    print('Конец программы')
```

Результаты четырёх запусков

```
n = 9043
Конец программы
===== RESTART =====
n = 789
sm = 24
===== RESTART =====
n = 70
pr = 7
===== RESTART =====
n = 54
pr = 20
```

Примечание: обе цифры в двузначном числе не могут быть равны нулю.

Пример 3.7. С клавиатуры вводится целое двузначное число. Если число неотрицательное, то увеличить каждую его цифру, отличную от 9, на 1. Если число отрицательное, то найти сумму его цифр.

Например, для числа 56 результат работы программы 67; для 59 — 69; для 99 — 99; для числа −67 ответ 13.

Программа

```
n = int(input('n >> '))
if n >= 0: # число неотрицательное
    n0 = n % 10
    n1 = n // 10
    if n0 != 9:
        n += 1
    if n1 != 9:
        n += 10
    print(f'{n = }')
else: # число отрицательное
    nn = abs(n)
    n0 = nn % 10
    n1 = nn // 10
    sm = n0 + n1
    print(f'{sm = }')
```

Результаты четырёх запусков

```
n >> 56
n = 67
===== RESTART =====
n >> 59
n = 69
===== RESTART =====
n >> 99
n = 99
===== RESTART =====
n >> -67
sm = 13
```

3.6 Тернарный оператор

Тернарный оператор используют для сокращения кода, а также для улучшения читаемости кода.

Условный оператор

```
if X:
    r = Y
else:
    r = Z
```

Тернарный оператор

```
r = Y if X else Z
```

В результате работы выполнится выражение Y, если значение X истинно, иначе выполнится выражение Z.



Рис. 7. Тернарный оператор

Поиск максимума

```
mx = x if x > y else y
```

Поиск max(x,y) и min(x,y)

```
x = int(input())
y = int(input())

mx, mn = (x, y) if x > y else (y, x)

print(f'{mn = }')
print(f'{mx = }')
```

Проверка числа на чётность

```
print('Чётное' if k % 2 == 0 else 'Нечётное')
```

В составе арифметического выражения

```
x = ...
y = x + (10 if x % 3 != 0 else -10)
print(y)
```

Компьютерный практикум



Темы: простые условия, операторы для организации ветвлений.

Задание: написать программы для решения задач 1–12.

Ограничения: циклы не использовать; составные логические выражения не использовать.

Примечания: Не забывайте, что целое число может быть отрицательным. Натуральные числа принадлежат интервалу: $[1, +\infty)$.

№ 1 (→). С клавиатуры вводится натуральное число. Проверить, является ли оно кратным трём. Ответ печатать в виде «Да» или «Нет».

№ 2 (→). Найти среди заданных целых чисел a, b, c, d количество чисел, равных нулю.

№ 3 (→). Найти среди заданных действительных чисел a, b, c, d количество положительных и количество отрицательных чисел.

№ 4 (→). Даны целые числа a, b, c . Найти $\max(a + b + c; abc) + \min(a^2 + b; b^2 + c)$.

№ 5 (→). С клавиатуры вводится целое число. Если, что оно трёхзначное, то найти сумму цифр числа. В противном случае ничего не делать.

Пример 1 работы программы

```
>>> n = 561
12
```

Пример 2 работы программы

```
>>> n = -104
5
```

№ 6 (→). С клавиатуры вводятся два натуральных числа. Если они разные, то найти их сумму, иначе — произведение.

Пример 1 работы программы

```
>>> n = 56
>>> m = 2
58
```

Пример 2 работы программы

```
>>> n = 5
>>> m = 5
25
```

Указания. Задачу решите с использованием тернарного оператора.

№ 7 (→). Дано целое число $n = 500$. Пользователю предлагается ввести основание системы счисления p для перевода числа n . Перевести число в одну из стандартных с.с. (т.е. $p = 2, 8, 16$) или вывести текст «Неизвестная с.с.».

Пример 1 работы программы

```
>>> p = 6
Неизвестная с.с.
```

Пример 2 работы программы

```
>>> p = 2
500_2 = '0b111110100'
```

№ 8 (→). Найти корни квадратного уравнения $ax^2 + bx + c = 0$; заданного коэффициентами a, b и c ($a \neq 0$). Использовать следующий алгоритм:

если $D < 0$, то выдать сообщение «Корни мнимые»;

если $D = 0$, то вычислить корень и выдать результат в виде « $x = \dots$ »;

если $D > 0$, то вычислить корни и выдать результат в виде « $x_1 = \dots, x_2 = \dots$ ».

Указания. Подобрать коэффициенты квадратного уравнения так, чтобы можно было полностью протестировать программу.

№ 9 (→). Определить является ли введённое целое число двузначным. Если является, то определить последнюю цифру числа, в противном случае вывести слово No.

Указания. Задачу решить в двух вариантах: 1) с использованием оператора if-else; 2) с использованием тернарного оператора.

№ 10 (→). Даны целые числа a, b, c . Найти 1) $\max(a + b + c; abc)$; 2) $\min(a^2 + b; b^2 + c)$.

Указания. Задания выполнить с использованием тернарного оператора.

№ 11 (→). С клавиатуры вводится натуральное число n . Проверьте, что последние цифры восьмеричной и четверичной записей этого числа совпадают. Ответ печатать в виде «Да» или «Нет».

Например, число 34 оканчивается в 8 с.с. на 2, и в 4 с.с. тоже.

Например, число 37 оканчивается в 8 с.с. на 5, а в 4 с.с. — на 1

Указания. Задачу решите с использованием тернарного оператора.

№ 12 (→). С клавиатуры вводятся целые координаты точек $A(x_1; y_1)$ и $B(x_2; y_2)$. Проверьте, что расстояние между точками A и B на плоскости меньше 12.

Указания. Задачу решите с использованием тернарного оператора.

Подсказки к решениям

Решение задачи № 1:

```
n = int(input('n = '))
if ...:
    print('Да')
else:
    print('Нет')
```

Решение задачи № 2:

```
a = int(input('a = '))
b = ...
...

cnt = 0
if a == 0:
    cnt += 1
if b == 0:
    cnt += 1
...
print(cnt)
```

Решение задачи № 3:

```
a = float(input('a = '))
b = ...
...

cntP = 0
cntN = 0
if a > 0:
    cntP += 1
elif a < 0:
    cntN += 1
...

print(f'{cntP = }, f'{cntN = }')
```

Решение задачи № 4:

```
...
v1 = a + b + c
v2 = a*b*c
v3 = ...
v4 = ...

if v1 > v2:
    mx = ...
else:
    mx = ...

if ...:
    mn = ...
else:
    mn = ...

print(mn + mx)
```

Решение задачи № 5:

```
n = int(input('n = '))
abs_n = abs(n)
if 100 <= abs_n <= 999:
    n0 = abs_n % 10
    n1 = ...
    n2 = ...
    sm = n0 + n1 + n2
    print(sm)
```

Решение задачи № 6:

```
...
res = ... if m != n else ...
print(res)
```

Решение задачи № 7:

```
n = ...
p = int(input('p = '))
if p == 2:
    print(f'{n}_{p} = ', ...)
elif ...:
    print(f'{n}_{p} = ', ...)
elif ...:
    print(f'{n}_{p} = ',...)
else:
    print('Неизвестная с.с.')
```

Решение задачи № 10:

```
a = ...
...
v1 = a + b + c
v2 = a*b*c
v3 = a**2 + b
v4 = b**2 + c

mx = ...
mn = ...
print(mx, mn)
```

Решение задачи № 8:

```
...

d = ...
if d < 0:
    print('Корни мнимые')
elif d == 0:
    x = ...
    print('x =', x)
else:
    x1 = ...
    x2 = ...
    print(...)
```

Решение задачи № 11:

```
n = int(input('n = '))
print(...)
```

Решение задачи № 12:

```
x1 = int(input('x1 = '))
y1 = int(input('y1 = '))
...

dist = ...
print(...)
```

Решение задачи № 9:

```
# Условный оператор
n = int(input('n = '))
nn = ...
if ...:
    n0 = ...
    print(n0)
else:
    print('No')

# Тернарный оператор
n = int(input('n = '))
nn = ...
print(... if ... else ...)
```

Ответы и решения

Решение задачи № 1:

```
n = int(input('n = '))
if n % 3 == 0:
    print('Да')
else:
    print('Нет')
```

Решение задачи № 2:

```
a = int(input('a = '))
b = int(input('b = '))
c = int(input('c = '))
d = int(input('d = '))

cnt = 0
if a == 0:
    cnt += 1
if b == 0:
    cnt += 1
if c == 0:
    cnt += 1
if d == 0:
    cnt += 1
print(cnt)
```

Решение задачи № 3:

```
a = float(input('a = '))
b = float(input('b = '))
c = float(input('c = '))
d = float(input('d = '))

cntP = 0 # количество положительных
cntN = 0 # количество отрицательных
if a > 0:
    cntP += 1
elif a < 0:
    cntN += 1

if b > 0:
    cntP += 1
elif b < 0:
    cntN += 1

if c > 0:
    cntP += 1
elif c < 0:
    cntN += 1

if d > 0:
    cntP += 1
elif d < 0:
    cntN += 1

print(f'{cntP = }', f'{cntN = }')
```

Решение задачи № 4:

```
a = int(input('a = '))
b = int(input('b = '))
c = int(input('c = '))
```

```
v1 = a + b + c
v2 = a*b*c
v3 = a**2 + b
v4 = b**2 + c
```

```
if v1 > v2:
    mx = v1
else:
    mx = v2
```

```
if v3 < v4:
    mn = v3
else:
    mn = v4
```

```
print(mn + mx)
```

Решение задачи № 5:

```
n = int(input('n = '))
abs_n = abs(n)
if 100 <= abs_n <= 999:
    n0 = abs_n % 10
    n1 = abs_n // 10 % 10
    n2 = abs_n // 100
    sm = n0 + n1 + n2
    print(sm)
```

Решение задачи № 6:

```
n = int(input('n = '))
m = int(input('m = '))
res = m + n if m != n else m * n
print(res)
```

Решение задачи № 7:

```
n = 500
p = int(input('p = '))
if p == 2:
    print(f'{n}_{p} =', bin(n))
elif p == 8:
    print(f'{n}_{p} =', oct(n))
elif p == 16:
    print(f'{n}_{p} =', hex(n))
else:
    print('Неизвестная с.с.')
```

Решение задачи № 8:

```
a = int(input('a = '))
b = int(input('b = '))
c = int(input('c = '))
```

```
d = b**2 - 4*a*c
if d < 0:
    print('Корни мнимые')
```

```
elif d == 0:
    x = -b / (2*a)
    print('x =', x)
```

```
else:
    x1 = (-b - d**0.5) / (2*a)
    x2 = (-b + d**0.5) / (2*a)
    print('x1 =', x1, 'x2 =', x2)
```

Решение задачи № 9:

```
# Условный оператор
n = int(input('n = '))
nn = abs(n)
if 10 <= nn <= 99:
    n0 = nn % 10
    print(n0)
else:
    print('No')

# Тернарный оператор
n = int(input('n = '))
nn = abs(n)
print(nn % 10 if 10 <= nn <= 99 else 'No')
```

Решение задачи № 10:

```
a = int(input('a = '))
b = int(input('b = '))
c = int(input('c = '))
v1 = a + b + c
v2 = a*b*c
v3 = a**2 + b
v4 = b**2 + c

mx = v1 if v1 > v2 else v2
mn = v3 if v3 < v4 else v3
print(mx, mn)
```

Решение задачи № 11:

```
n = int(input('n = '))
print('Да' if n % 8 == n % 4 else 'Нет')
```

Решение задачи № 12:

```
x1 = int(input('x1 = '))
y1 = int(input('y1 = '))

x2 = int(input('x2 = '))
y2 = int(input('y2 = '))

dist = ((x1 - x2)**2 + (y1 - y2)**2)**0.5
print(dist)

print('Да' if dist < 12 else 'Нет')
```



Метод утёнка — психологический метод решения задачи за счёт делегирования её мысленно-му помощнику. Если ваш код не работает или работает неправильно, то объясните свой код утёнку (пошаговый разбор кода поможет найти ошибки).

Готовимся к ЕГЭ вместе с «Отличным кодом»! (очно или онлайн)

Проводится набор в **новые** мини-группы (очно не более 6 человек + онлайн)

Информатика 10 класс; 1 занятие в неделю (180 минут); старт в начале сентября

Информатика 11 класс; 1 занятие в неделю (180 минут); старт в начале сентября

Математика 10 класс; 11 класс: 1 занятие в неделю (180 минут); старт в начале сентября

Что ожидает учеников:

- опытные преподаватели (кандидаты физ.-мат. наук);
- авторские материалы для подготовки;
- необходимая база по всем заданиям ЕГЭ по предмету;
- домашние работы с индивидуальной проверкой и разбором проблем;
- регулярные пробники для оценки уровня знаний;
- возможность онлайн-консультаций во внеурочное время.

Вопросы; запись



#Python #ОтличныйКод #ЕГЭИнформатика #ПрограммированиеДляШкольников #ПодготовкаКЕГЭ #ОнлайнОбучение

Большое спасибо за подготовку! Узнала столько всего нового, даже не верится, что всего год назад с компьютером совсем работать не умела 🙏 Очень рада, что попала на интенсивы к Вам, непередаваемая атмосфера комфорта! Было очень интересно и здорово ❤️

Информатика (КЕГЭ) 19.06.2026
Ваш первичный балл: 22
Ваш тестовый балл: **83**

12:05

Спасибо вам большое 🙏

Информатика (КЕГЭ) 19.06.2026
Ваш первичный балл: 24
Ваш тестовый балл: **88**

Информатика				
Полученные баллы				
95				
Дата экзамена				
18.06.2026				
Здравствуй Елен Владимировна, 95 🙏❤️				
10	34	Цифры	1	1
11	256	Цифры	1	1
12	1001	Цифры	0	1
13	381	Цифры	1	1
14	1875	Цифры	1	1
15	18	Цифры	1	1
16	41932098	Цифры	1	1
17	Подробнее	Матрица	1	1
18	Подробнее	Матрица	1	1
19	16	Цифры	1	1
20	Подробнее	Матрица	1	1
21	45	Цифры	1	1
22	6	Цифры	1	1
23	35	Цифры	1	1
24	155	Цифры	0	1
25	Подробнее	Матрица	1	1
26	Подробнее	Матрица	2	2
27	Подробнее	Матрица	2	2
Итого			27	29

* Первичный балл

если бы не Вы, такого результата не было бы. спасибо Вам огромное!!!

Информатика (КЕГЭ) 19.06.2026
Ваш первичный балл: 23
Ваш тестовый балл: **85**

Доброе утро, у меня 90 баллов 08:47

№	Ваш ответ	Допустимые символы		
1	15	Цифры		
2	zxxy	Латиница, дефисы		
3	1629146	Цифры		
4	22	Цифры		
5	17	Цифры		
6	110	Цифры		
7	31875	Цифры	1	1
8	46549	Цифры	0	1
9	14	Цифры	1	1
10	21	Цифры	1	1
11	9	Цифры	1	1
12	6124	Цифры	1	1
13	752	Цифры	1	1
14	2001	Цифры	1	1
15	448	Цифры	1	1
16	297717768	Цифры	1	1
17	Подробнее	Матрица	1	1
18	Подробнее	Матрица	1	1
19	37	Цифры	1	1
20	Подробнее	Матрица	1	1
21	59	Цифры	1	1
22	19	Цифры	1	1
23	704	Цифры	1	1
24	2386	Цифры	1	1
25	Подробнее	Матрица	0	1
26	Подробнее	Матрица	2	2
27	Подробнее	Матрица	0	2
Спасибо! 08:59			25	29



Дорогая Елена Владимировна! Огромнейшее Вам спасибо за подготовку Максима! Сегодня получил результат - 95 баллов. Если бы не Вы, Ваше внимание к нему (и его ошибкам 🙏), Ваша забота и требовательность, то этого всего не было бы. От всей души - спасибо!!! 18:08

Доброе утро, Елена Владимировна! У Андрея 90 баллов. Спасибо вам огромнейшее! 8:10

ГОТОВИМСЯ К ЕГЭ ПО МАТЕМАТИКЕ

Не откладывая успех на потом!

Хороший результат складывается не за месяц, а шаг за шагом.

Начни подготовку сейчас — поблагодари себя в мае!

Цель: 80+ баллов!

На занятиях ученики получают:

- Понятные объяснения даже самых сложных тем
- Регулярную практику и разбор ошибок
- Контроль прогресса на каждом этапе подготовки
- Поддержку преподавателя и ответы на все вопросы
- Чёткую цель — 80+ баллов и уверенность на экзамене

Для учеников — уверенность. Каждая решённая задача приближает к цели и даёт ощущение: «Я умею решать. Я справлюсь!»

Для родителей — спокойствие. Вы знаете:

- ребёнок занимается регулярно;
- ничего не упускает;
- материал понимает;
- становится сильнее с каждым днём.

 Подготовка идёт по плану!

Сейчас идёт набор в новые группы! Начни подготовку спокойно, системно и без постоянного стресса. Напишите нам в MAX

Шаг за шагом — к высоким баллам на ЕГЭ!

ОТЛИЧНЫЙ КОД ПРОЕКТ

Подготовка к ЕГЭ по математике на Python и профильной математике

ГОТОВИМСЯ К ЕГЭ ПО ИНФОРМАТИКЕ

Информатика — это логика, которую можно понять!

Мы учимся понимать алгоритмы и писать код, а не зубрим решения!

Системная подготовка — уверенность на экзамене!

Цель: 80+ баллов!

На занятиях ученики получают:

- Понятные объяснения даже самых сложных тем
- Регулярную практику и разбор ошибок
- Контроль прогресса на каждом этапе подготовки
- Поддержку преподавателя и ответы на все вопросы
- Чёткую цель — 80+ баллов и уверенность на экзамене

Для учеников — уверенность. Каждая решённая задача приближает к цели и даёт ощущение: «Я понимаю, что делаю. Я справлюсь!»

Для родителей — спокойствие. Вы знаете:

- ребёнок занимается регулярно;
- ничего не упускает;
- материал понимает;
- становится сильнее с каждым днём.

 Подготовка идёт по плану!

Сейчас идёт набор в новые группы! Начни подготовку спокойно, системно и без постоянного стресса. Напишите нам в MAX

Шаг за шагом — к высоким баллам на ЕГЭ!

ОТЛИЧНЫЙ КОД ПРОЕКТ

Подготовка к ЕГЭ по информатике на Python и профильной математике

«Нулевой курс мехмата» ЮФУ, 2026-27 уч. год



Программы «Нулевого курса» входят в полный цикл подготовки к ЕГЭ по математике и информатике



Занятия проводятся еженедельно с октября по апрель



Опытные преподаватели мехмата ЮФУ, кандидаты и доктора наук



Имеются группы онлайн-подготовки



Наши выпускники учатся в лучших вузах России



Запись в группы с апреля по сентябрь по адресу: ZeroCourseMM@mail.ru



mmcs.sfedu.ru/zerocourse
t.me/ZeroCourseMM
 Ростов-на-Дону, ул. Мильчакова, 8а, ауд. 117
Организационное собрание:
27 сентября 2026, 10.00, ауд. 120

