

Алгоритмизация и программирование

Решение задач на компьютере

Условный оператор

Задача №1. Найти большее из 3-х целых чисел

Вариант

№1

```
a=int(input())
b=int(input())
c=int(input())
if (a>=b)and(a>=c):
    print(a)
elif (b>=a)and(b>=c):
    print(b)
elif (c>=a)and(c>=b):
    print(c)
```

Вариант

№2

```
a=int(input())
b=int(input())
c=int(input())
max = a
if max < b:
    max=b
if max < c:
    max = c
print(max)
```

Вариант

№3

```
a=int(input())
b=int(input())
c=int(input())
if a >= b >= c:
    print (a)
elif b >= a >= c:
    print (b)
else:
    print (c)
```

Задач

а С клавиатуры вводится значение целой переменной a . Если это **трёхзначное число**, которое **делится на 7**, то вывести сообщение «Да», в противном случае «Нет»

№2

Для этого нужно, чтобы одновременно выполнились **три условия**:

- 1) число не меньше 100;
- 2) число меньше 1000;
- 3) число делится на 7, то есть остаток от его деления на 7 равен нулю.

В условном операторе эти три простых условия должны быть связаны с помощью двух операций «И»:

```
if 100 <= a and a < 1000 and a % 7 == 0:  
    print( "Да!" )  
else:  
    print( "Нет." )
```

Проверка условий

Число a кратно X

```
if a % X == 0:
```

```
    print ("кратно X")
```

Число a НЕ кратно X

```
if a % X != 0:
```

```
    print ("НЕ кратно X")
```

Число a – четное

```
if a % 2 == 0:
```

```
    print ("четное")
```

Число a - НЕчетное

```
if a % 2 != 0:
```

```
    print ("НЕчетное")
```

Самостоятельная практическая работа

Задача №1 - 2
балла

Домашнее задание

- записи в тетради,
- **задача №2 на карточке – написать программу,**
- задача №3 на карточке – по желанию

Повторение:

Тема: Программирование Python (Питон)

Программирование — это создание программ для компьютеров. Этим занимаются программисты.

Комментарий – текст-пояснение в коде программы, не обрабатывается.

Вывод на экран

Команда (оператор)	Видим на экране	Комментарий
<pre>print("Привет, ") print("Вася!")</pre>	Привет, Вася!	# текст пишем в кавычках, переход на новую строку автоматически
<pre>print("Привет, " , end="") print("Вася!")</pre>	Привет, Вася!	# курсор останется в той же строке
<pre>c = 12 print(c)</pre>	12	# вывод значения переменной c

Переменная — это величина, которая имеет

Имя

Тип

Значение

<p>МОЖНО использовать</p> <ul style="list-style-type: none">• латинские буквы (A-Z, a-z)• цифры (не 1-ая в имени!)• знак подчеркивания _ <p>НЕЛЬЗЯ использовать скобки, знаки ", &, , *, +, -, !, ? и др</p> <p>Например: a, sum, S_2, Massal</p>	<ul style="list-style-type: none">• int – целочисленный (1, -78, 1089)• float – вещественный (1.8, -56.904, 1009070)• str – строковый (text, F, a)• bool – логический (true, false)	<p>a = 5</p> <p>= оператор присваивания</p>
--	--	---

Ввод с клавиатуры



a = input()

1. Программа ждет, пока пользователь введет значение и нажмет *Enter*.
2. Введенное значение записывается в переменную a.

Задача. Ввести два числа и найти их сумму.

```
a = int( input() ) # ввести 1-ое число целого типа
b = int( input() ) # ввести 2-ое число целого типа
c = a + b          # вычислить их сумму
print ( c )       # вывести сумму на экран
```

Урок №2. Тема: Программирование Python. Арифметические операции.

Арифметические выражения

$$a \leftarrow \frac{c + b - 1}{2} \cdot d$$

Линейная запись (в одну строку):

$$a = (c + b - 1) / 2 * d$$

Операции: + -

* - умножение

/ - деление

** - возведение в степень ($x^2 \rightarrow x**2$)

Порядок выполнения операций

$$a = (c + b**5*3 - 1) / 2 * d$$

Приоритет (старшинство):

- 1) скобки
- 2) возведение в степень **
- 3) умножение и деление
- 4) сложение и вычитание

$$a = \frac{c + b^5 \cdot 3 - 1}{2} \cdot d$$

Арифметическая операция	Название операции	Пример (где b = 9, c = 2)	Ответ
+	сложение	x1 = b + c print (x1)	11
-	вычитание	x2 = b - c print (x2)	7
*	умножение	x3 = b * c print (x3)	18
**	возведение в степень	x4 = b ** c print (x4)	81
/	деление	x5 = b / c print (x5)	4,5
//	деление нацело (остаток отбрасывается)	x6 = b // c print (x6)	4
%	остаток от деления	x7 = b % c print (x7)	1

$$\begin{array}{r}
 9 \overline{) 2} \\
 \underline{8} \\
 1
 \end{array}$$

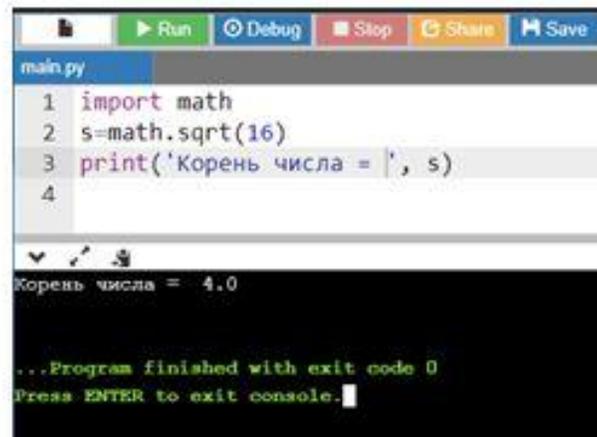
4 ← целая часть после деления

↑
остаток от деления

Модуль math- модуль предоставляет обширный функционал для работы с числами.

```
import math
```

Вычисления квадратного корня числа: `math.sqrt(X)` - квадратный корень из X.



```
main.py
1 import math
2 s=math.sqrt(16)
3 print('Корень числа = ', s)
4
```

Корень числа = 4.0

...Program finished with exit code 0
Press ENTER to exit console.

Подключили модуль math

Операция вычисления квадратного корня из 16

Вывод результата на экран

Другие математические операции:

`math.ceil(X)` – округление до ближайшего большего числа.

`math.fabs(X)` - модуль X.

`math.factorial(X)` - факториал числа X.

`math.floor(X)` - округление вниз.

`math.fmod(X, Y)` - остаток от деления X на Y.

`math.frexp(X)` - возвращает мантиссу и экспоненту числа.

`math.fsum(последовательность)` - сумма всех членов последовательности. Эквивалент встроенной функции `sum()`, но

`math.fsum()` более точна для чисел с плавающей точкой.

`math.isfinite(X)` - является ли X числом.

`math.trunc(X)` - отсекает значение X до целого.

`math.exp(X)` - e^X .

`math.log(X, [base])` - логарифм X по основанию base. Если base не указан, вычисляется натуральный логарифм.

`math.log1p(X)` - натуральный логарифм (1 + X). При $X \rightarrow 0$ точнее, чем `math.log(1+X)`.

`math.log10(X)` - логарифм X по основанию 10.

`math.log2(X)` - логарифм X по основанию 2.

Урок №3. Тема: Python. Алгоритмы с ветвлением. Условный оператор.

Ветвление – это команда алгоритма, в которой делается выбор, выполнять или не выполнять какую-нибудь группу команд в зависимости от условий.

Форма ветвления	Синтаксис	Пример	
		Блок-схема	Программа
ПОЛНАЯ	if Условие: Блок инструкций 1 else: Блок инструкций 2	<pre> graph TD Start([начало]) --> Input[/ВВОД a, b/] Input --> M_a[M = a] M_a --> Decision{b > a?} Decision -- да --> M_b[M = b] Decision -- нет --> Output[/Вывод M/] M_b --> Output Output --> End([конец]) </pre>	<pre> print("Введите два целых числа") a = int(input()) b = int(input()) if a > b: M = a else: M = b print("Наибольшее число", M) </pre> <p>полная форма условного оператора</p> <p>отступы</p>
НЕПОЛНАЯ	if Условие: Блок инструкций 1	<pre> graph TD Start([начало]) --> Input[/ВВОД a, b/] Input --> M_a[M = a] M_a --> Decision{b > a?} Decision -- да --> M_b[M = b] Decision -- нет --> Output[/Вывод M/] M_b --> Output Output --> End([конец]) </pre>	<pre> print("Введите два целых числа") a = int(input()) b = int(input()) M = a if b > a: M = b print("Наибольшее число", M) </pre> <p>неполная форма условного оператора</p>

Операторы сравнения: < — меньше; > — больше; <= — меньше или равно; >= — больше или равно; == — равно; != — не равно.

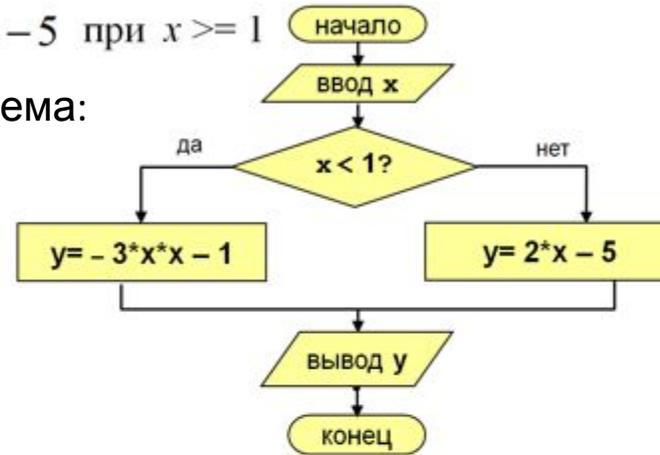
Примеры условий: cost < 1000; x == y; m >= 17; 1 <= x <= 10

Пример задачи с условием:

Для данного значения x вычислить значение функции:

$$y = \begin{cases} -3x^2 - 1 & \text{при } x < 1 \\ 2x - 5 & \text{при } x \geq 1 \end{cases}$$

Блок-схема:



```
main.py
1
2 print ('Введите число x = ')
3 x=int(input())
4 if x < 1:
5     y=-3*x*x-1
6 else:
7     y=2*x-5
8 print('Результат =', y)
9
```

Powered by trinket

Введите число x =
6
('Результат =', 7)

Ввели неизвестное x с клавиатуры

Если $x < 1$, то $y = -3x^2 - 1$

Иначе $y = 2x - 5$

Вывод результат y на экран