

# Знакомство с языком Python (семинары)

## Задание 1. Язык математики

В первый же день на сайте отвалилась формула по расчёту рекламной метрики, и только Вася может её поправить. Часть программы с вводными данными представлена ниже, отдельно записана формула на математическом языке.

Дана программа:

`a = 8`

`b = 10`

`c = 12`

`d = 18`

Продолжите программу: переведите выражение с математического языка на язык Python, запишите его в переменную `res` и выведите результат.

Выражение:

$$\frac{(-3 + a^2) * b - 2^3}{c - 2 * d}$$

### Подсказка № 1

Прежде чем реализовать формулу в Python, разберитесь с порядком операций. В Python сначала выполняются операции возведения в степень, затем умножение и деление, а потом сложение и вычитание. Это соответствует математическому порядку операций.

### Подсказка № 2

Помните, что выражение в Python должно учитывать порядок операций. Скобки в математическом выражении помогают определить, какие операции следует выполнять первыми.

### Подсказка № 3

Для реализации формулы создайте переменные делимое и делитель. Реализуйте в них части формулы и поделите полученные значения.

**Эталонное решение:**

```
# Инициализация переменных с заданными значениями

a = 8

b = 10

c = 12

d = 18


# Вычисление числителя формулы

# - Сначала вычисляем возведение a в квадрат

# - Затем прибавляем -3

# - Умножаем результат на b

# - Вычисляем 2 в степени 3 и вычитаем его из предыдущего результата

dividend = ((-3 + a ** 2) * b - 2 ** 3)


# Вычисление знаменателя формулы

# - Умножаем 2 на d

# - Вычисляем разность c и полученного результата

divider = (c - 2 * d)


# Вычисление результата деления числителя на знаменатель

res = dividend / divider


# Вывод результата на экран

print(res)
```

## Задача 2. Часы

Напишите программу, которая получает на вход число  $n$  (количество минут), затем считает, сколько это будет в часах и сколько минут останется, и выводит на экран эти два результата.

### Подсказка № 1

Для получения количества часов из общего числа минут, вам нужно использовать целочисленное деление (`//`). Это позволит вам узнать, сколько полных часов содержится в данном числе минут.

### Подсказка № 2

Чтобы узнать, сколько минут останется после извлечения полных часов, используйте оператор остатка от деления (`%`). Это поможет вам найти остаток минут, которые не полностью заполняют час.

### Подсказка № 3

Перед выполнением вычислений убедитесь, что пользователь вводит корректное число минут. Вы можете использовать функцию `int()` для преобразования входного значения из строки в целое число.

### Эталонное решение:

```
# Ввод количества минут
minutes = int(input('Введите количество минут: '))

# Вычисление количества полных часов
hours = minutes // 60

# Вычисление оставшихся минут
rest_minutes = minutes % 60

# Вывод результатов
print('Часов:', hours)
print('Осталось минут:', rest_minutes)
```

### Задача 3. Счастливый билет

Вы пользуетесь общественным транспортом? Вероятно, вы расплачивались за проезд и получали билет с номером.

Счастливым билетом называют такой билет с шестизначным номером, где сумма первых трех цифр равна сумме последних трех.

Т.е. билет с номером 385916 – счастливый, т.к.  $3+8+5=9+1+6$ .

Вам требуется написать программу, которая проверяет счастливость билета с номером  $n$  и выводит на экран `yes` или `no`.

#### Подсказка № 1

Для того чтобы получить каждую цифру номера билета, используйте операции целочисленного деления (`//`) и остатка от деления (`%`). Эти операции помогут выделить каждую цифру в номере. Например, чтобы получить первую цифру шестизначного числа, нужно разделить его на 100000.

#### Подсказка № 2

После того как вы выделили каждую цифру, разделите их на две группы. Первые три цифры и последние три цифры. Найдите сумму цифр для каждой группы. Это можно сделать с помощью обычных арифметических операций.

#### Подсказка № 3

Сравните суммы первых и последних трех цифр. Если они равны, билет считается счастливым. В противном случае билет не счастливый. Для этого используйте условные операторы `if` и `else`.

#### Эталонное решение:

```
# Ввод номера билета

n = int(input('Введите номер билета: '))

# Разделение числа на отдельные цифры

n1 = n // 100000          # первая цифра

n2 = (n % 100000) // 10000 # вторая цифра

n3 = (n % 10000) // 1000   # третья цифра

n4 = (n % 1000) // 100     # четвертая цифра
```

```
n5 = (n % 100) // 10      # пятая цифра
n6 = n % 10              # шестая цифра

# Проверка на счастливость билета
if n1 + n2 + n3 == n4 + n5 + n6:
    print('yes')  # Вывод 'yes', если билет счастливый
else:
    print('no')   # Вывод 'no', если билет не счастливый
```

#### Задача 4. Площадь прямоугольного треугольника

Прямоугольный треугольник — это треугольник, один из углов которого является прямым (90 градусов). У прямоугольного треугольника два катета, которые образуют прямой угол, и гипотенуза, противоположная этому углу.

Вам требуется написать программу, которая запрашивает у пользователя длины двух катетов и выводит площадь треугольника.

Площадь прямоугольного треугольника вычисляется по формуле:

$$S = \frac{1}{2}ab$$

где  $a$  и  $b$  — длины катетов, а  $S$  — площадь.

##### Подсказка № 1

Прямоугольный треугольник — это треугольник с одним прямым углом. Два катета треугольника образуют этот угол, а третья сторона — гипотенуза. Для вычисления площади нужны только длины катетов.

##### Подсказка № 2

Для того чтобы получить значения катетов от пользователя, нужно использовать функцию `input()`. Эта функция запрашивает ввод данных с клавиатуры в виде

строки. Чтобы получить число, строку нужно преобразовать с помощью функции `int()`.

### Подсказка № 3

Не забывайте, что операция умножения в Python обозначается символом `*`, а деление — символом `/`. Так как в задаче требуется вычислить площадь, вам нужно использовать оба эти оператора.

### Эталонное решение:

```
# Запрос длин катетов у пользователя

leg_1 = int(input('Введите длину первого катета: '))

leg_2 = int(input('Введите длину второго катета: '))


# Вычисление площади по формуле

square = leg_1 * leg_2 / 2


# Вывод площади треугольника

print('Площадь прямоугольного треугольника:', square)
```

### Задача 5\*. Поменять местами: не всё так просто! (необязательная, повышенной сложности)

Напишите программу, которая меняла бы значения двух переменных местами, но без использования третьей переменной и синтаксического сахара, который мы разбирали, а именно: без конструкции `a, b = b, a`. В переменные будут вводиться только числа.

```
a = int(input('Введите первое число: '))
```

```
b = int(input('Введите второе число: '))
```

```
print(a, b)
```

```
# стереть эту строчку и вставить свой код здесь
```

```
print(a, b)
```

Изменять, удалять, менять местами первую, вторую, третью и последнюю строки нельзя. В четвертую строку можно вставлять сколько угодно кода, не трогая последний print.

#### Подсказка № 1

Для выполнения задачи без использования третьей переменной можно использовать арифметические операции. Вам пригодятся операции сложения и вычитания для обмена значениями.

#### Подсказка № 2

Сначала присвойте переменной **a** значение суммы **a** и **b**. Это обеспечит сохранение обоих исходных значений в **a**, поскольку **b** также входит в эту сумму.

#### Подсказка № 3

Далее присвойте переменной **b** разницу нового значения **a** и текущего значения **b**. Это действие вернет старое значение **a**, так как **a** уже содержит сумму обоих чисел.

#### Подсказка № 4

Завершите обмен значениями, присвоив переменной **a** разницу между новым значением **a** и обновленным значением **b**. Это действие вернет старое значение **b**.

#### Эталонное решение:

```
a = int(input('Введите первое число: '))
b = int(input('Введите второе число: '))
print(a, b)

# Меняем значения a и b без использования третьей переменной
a = a + b # Сначала a становится суммой a и b
b = a - b # Затем b становится старым значением a (a + b - b = a)
a = a - b # Наконец, a становится старым значением b (a + b - a = b)
print(a, b)
```