

Практика 3. Рекурсия, рекурсия, (древовидная) рекурсия!

Рекурсия

Вопрос 1

Распечатай обратный отсчет с помощью рекурсии.

```
def countdown(n):  
    """  
    >>> countdown(3)  
    3  
    2  
    1  
    """
```

Ответ

Вопрос 2

Существует ли простой способ изменить `countdown`, чтобы считать в прямом порядке?

Ответ

Вопрос 3

Напиши функцию `expt(base, power)`, которая возводит `base` в степень `power`. То есть `expt(3, 2)` возвращает 9, `expt(2, 3)` возвращает 8. Считай, что `power` всегда положительный. Используй рекурсию, а не `pow`.

```
def expt(base, power):
```

Ответ

Вопрос 4

Напиши реализацию функции `is_prime`, которая возвращает `True`, если `n` — простое число, в противном случае — `False`. Конечно требуется применить рекурсию!

```
def is_prime(n):
```

Ответ

Вопрос 5

Напиши функцию `sum_primes_up_to(n)`, которая суммирует все простые числа до `n` включительно. Нужно использовать только что созданную функцию `is_prime`.

```
def sum_primes_up_to(n):
```

Ответ

Вопрос 6

Нарисуй диаграмму окружения для следующего кода:

```
def rec(x, y):  
    if y > 0:  
        return x * rec(x, y - 1)  
    return 1  
rec(3, 2)
```

Кстати, что делает эта функция?

Ответ

Древовидная рекурсия

Вопрос 7

Треугольник Паскаля — бесконечная таблица значений, имеющая треугольную форму и задаваемая рекурсивно. Значения из треугольника Паскаля соответствуют коэффициентам разложения $(x+a)^n$. Каждый элемент треугольника имеет координату, задаваемую через номер строки и номер элемента в строке. Каждый элемент треугольника Паскаля определяется суммой элементов, находящегося над рассматриваемым и левым соседом последнего. Вместо значения недостающих элементов используется 0. Ниже представлено несколько строк треугольника Паскаля.

Элемент:	0	1	2	3	4	5
Строка 0:	1					
Строка 1:	1	1				
Строка 2:	1	2	1			
Строка 3:	1	3	3	1		
Строка 4:	1	4	6	4	1	
Строка 5:	1	5	10	10	5	1

Напиши функцию `pascal(row, column)`, которая принимает строку и столбец и возвращает значение соответствующего элемента треугольника Паскаля. Нужно использовать рекурсию.

```
def pascal(row, column):
```

Ответ

Вопрос 8

Преподавателю потребовалось распечатать одностраничный раздаточный материал для студентов. Однако по непонятным причинам оба его принтера сломались. Первый принтер может печатать только по `n1` копий, второй принтер может печатать только по `n2` копий. Помогите преподавателю понять, сможет ли он распечатать в точности необходимое количество раздаточного материала!

Попробуй решить и без использования, и с использованием вспомогательной внутренней функции.

```
def has_sum(sum, n1, n2):
    """
    >>> has_sum(1, 3, 5)
    False
    >>> has_sum(5, 3, 5) # 1(5) + 0(3) = 5
    True
    >>> has_sum(11, 3, 5) # 2(3) + 1(5) = 11
    True
    """
```

Ответ

Вопрос 9*

На следующий день принтерам совсем поплохело! Теперь при каждом использовании принтер А печатает случайное количество копий x от 50 до 60, а принтер В выдает случайное количество копий y от 130 до 140. Преподаватель в этой ситуации понизил свои ожидания и будет доволен, если распечатается не менее `lower` копий, но строго не более `upper` копий.

```
def sum_range(lower, upper):
    """
    >>> sum_range(45, 60) # принтер А попадет в диапазон
    True
    >>> sum_range(40, 55) # принтер А может выйти за диапазон
    False
    >>> sum_range(170, 201) # оба принтера напечатают 180-200 копий
    True
    """
```

Ответ

