

1. В Java есть единственный оператор, способный обеспечить остаток от операции деления. Два числа передаются в качестве параметров. Первый параметр, разделенный на второй параметр, будет иметь остаток, возможно, ноль. Верните это значение.

Пример:

```
remainder(1, 3) → 1  
remainder(3, 4) → 3  
remainder(-9, 45) → -9  
remainder(5, 5) → 0
```

2. Напишите функцию, которая принимает основание и высоту треугольника и возвращает его площадь.

Пример:

```
triArea(3, 2) → 3  
triArea(7, 4) → 14  
triArea(10, 10) → 50
```

3. В этой задаче фермер просит вас сказать ему, сколько ног можно сосчитать среди всех его животных. Фермер разводит три вида:

chickens = 2 legs

cows = 4 legs

pigs = 4 legs

Фермер подсчитал своих животных, и он дает вам промежуточный итог для каждого вида. Вы должны реализовать функцию, которая возвращает общее количество ног всех животных.

Пример:

```
animals(2, 3, 5) → 36  
animals(1, 2, 3) → 22  
animals(5, 2, 8) → 50
```

4. Создайте функцию, которая принимает три аргумента (prob, prize, pay) и возвращает true, если $\text{prob} * \text{prize} > \text{pay}$; в противном случае возвращает false.

Чтобы проиллюстрировать это: profitableGamble (0,2, 50, 9) должен выдать значение true, поскольку $1 (0,2 * 50 - 9)$, а $1 > 0$.

Пример:

```
profitableGamble(0.2, 50, 9) → true  
profitableGamble(0.9, 1, 2) → false
```

```
profitableGamble(0.9, 3, 2) → true
```

5. Напишите функцию, которая принимает 3 числа и возвращает, что нужно сделать с а и b: они должны быть сложены, вычитаны, умножены или разделены, чтобы получить N. Если ни одна из операций не может дать N, верните "none".
3 числа – это N, а и b.

Пример:

```
operation(24, 15, 9) → "added"
```

```
operation(24, 26, 2) → "subtracted"
```

```
operation(15, 11, 11) → "none"
```

Примечания:

- В качестве тестового ввода используются только целые числа.
- Числа должны быть добавлены, вычтены, разделены или умножены в порядке их появления в параметрах.

6. Создайте функцию, которая возвращает значение ASCII переданного символа.

Пример:

```
ctoa('A') → 65
```

```
ctoa('m') → 109
```

```
ctoa('[') → 91
```

```
ctoa('\')
```

7. Напишите функцию, которая берет последнее число из последовательного списка чисел и возвращает сумму всех чисел до него и включая его.

Пример:

```
addUpTo(3) → 6  
// 1 + 2 + 3 = 6
```

```
addUpTo(10) → 55  
// 1 + 2 + 3 + ... + 10 = 55
```

```
addUpTo(7) → 28  
// 1 + 2 + 3 + ... + 7 = 28
```

8. Создайте функцию, которая находит максимальное значение третьего ребра треугольника, где длины сторон являются целыми числами.

Пример:

```
nextEdge(8, 10) → 17
```

```
nextEdge(5, 7) → 11
```

```
nextEdge(9, 2) → 10
```

9. Создайте функцию, которая принимает массив чисел и возвращает сумму его кубов.

Пример:

```
sumOfCubes([1, 5, 9]) → 855  
// Since  $1^3 + 5^3 + 9^3 = 1 + 125 + 729 = 855$ 
```

```
sumOfCubes([3, 4, 5]) → 216
```

```
sumOfCubes([2]) → 8
```

```
sumOfCubes([]) → 0
```

10. Создайте функцию, которая будет для данного a, b, c выполнять следующие действия:

- Добавьте A к себе B раз.
- Проверьте, делится ли результат на C.

Пример:

```
abcmath(42, 5, 10) → false  
//  $42+42 = 84, 84+84 = 168, 168+168 = 336, 336+336 = 672, 672+672 = 1344$   
// 1344 is not divisible by 10
```

```
abcmath(5, 2, 1) → true
```

```
abcmath(1, 2, 3) → false
```