

Рассмотрим четыре реализации одной и той же функции на языке программирования *python*. Определите, что должна вычислять функция. Какие из реализаций работают корректно?

(a)

```
def solve(n, k):
    if n < 0 or k < 0 or k > n: return 0
    if n == 0 or k == 0 or n == k: return 1
    s = 0
    step = k
    if n > 47: step = 10
    for i in range(n + 1):
        s += solve(n - step, i * solve(step, k - i))
    return s
```

(b)

```
def solve(n, k):
    A = [ 0 for i in range(n+1) ]
    for s in range(16**n)
        tmp = s
        odd = 0
        for t in range(n):
            if tmp % 2: odd += 1
            tmp = tmp // 16
        A[odd] += 1
    return A[k] // 2**(3*n)
```

(c)

```
def solve(n, k):
    if k == 0 or n == k: return 1
    return solve(n + 1, k) - solve(n, k - 1)
```

(d)

```
def solve(n, k):
    if k == 0 or n == k: return 1
    return solve(n, k + 1) * (k + 1) // (n - k)
```

*Замечание.* Некоторые разъяснения к синтаксису *python*.

- `range(x)` возвращает массив `[0,1,..., x-1]`
- `**` возведение в степень, например, `2**5 == 32`
- `%` взятие остатка от деления, например, `7 % 3 == 1`
- `//` целочисленное деление, например, `7 // 3 == 2`