

Лекция № 9

Т. Ф. Хирьянов

Рекурсия

```
def make_matroska(size, number):  
    if number > 1:  
        print('низ размера', size)  
        make_matroska(size - 1, number - 1)  
        print('верх размера', size)  
    else:  
        print('матрешечка размера', size)
```

Поиск НОД. Алгоритм Евклида

```
def my_gcd(a, b):  
    if b == 0:  
        return a  
    else:  
        return my_gcd(b, a%b)
```

Факториал

```
def factor(n):  
    return 1 if n == 0 else factor(n - 1)*n
```

Числа Фибоначчи

```
def fib(n):  
    if n < 2:  
        return n  
    else:  
        return fib(n - 1) + fib(n - 2)
```

Быстрое возведение в степень

```
def fast_power(a, n):  
    if n == 0:  
        return 1  
    elif n%2 == 1:
```

```

        return a*fast_power(a, n - 1)
    else:
        n%2 == 0
        return fast_power(a*a, n//2)

```

Ханойские башни

```

def hanoi(n, i=1, j=2):
    if n == 1:
        print('переставить 1', 'блин с', i, 'на', j, 'стержень')
    else:
        hanoi(n - 1, i, 6 - i - j)
        print('переставить', n, 'блин', n - 1, i, 'на', j, 'стержень')
        hanoi(n - 1, 6 - i - j, j)

```

Генерация комбинаторных объектов

```

def bin_gen(n, prefix = ""):
    if n == 0:
        print(prefix)
    else:
        assert(n > 0)
        bin_gen(n - 1, prefix + '0')
        bin_gen(n - 1, prefix + '1')

def perestanov_gen(n, A=[])
    if len(A) == n:
        print(*A)
    else:
        for x in range(1, n + 1):
            if x not in A:
                perestanov_gen(n, A + [x])

```