

# Как измерять время работы функций.

Вашему вниманию предлагаются несколько программ с функциями.  
Функция `f_sl()` ([ссылка для скачивания](#)):

```
def f_sl(n, MOD):
    if n < 3:
        return 1
    return (f_sl(n - 2, MOD) + f_sl(n - 1, MOD)) % MOD

MOD = 1000000009
k = 10
print(f_sl(k, MOD))
```

Функция `f_fa()` ([ссылка для скачивания](#)):

```
def f_fa(n, MOD):
    f1, f2 = 1, 1
    for k in range(3, n + 1):
        f1, f2 = f2, (f1 + f2) % MOD
    return f2

MOD = 1000000009
k = 10
print(f_fa(k, MOD))
```

И набор функций `mu()`, `sq()`, `pw()` ([ссылка для скачивания](#)):

```
def mu(
    a11, a12, a21, a22,
    b11, b12, b21, b22,
    MOD
):
    return (a11 * b11 + a12 * b21) % MOD, \
           (a11 * b12 + a12 * b22) % MOD, \
           (a21 * b11 + a22 * b21) % MOD, \
           (a21 * b12 + a22 * b22) % MOD

def sq(a11, a12, a21, a22, MOD):
    return (a11 * a11 + a21 * a12) % MOD, \
           (a11 * a12 + a12 * a22) % MOD, \
           (a21 * a11 + a22 * a21) % MOD, \
           (a21 * a12 + a22 * a22) % MOD

def pw(a11, a12, a21, a22, n, MOD):
    if n == 0:
        return 1, 0, 0, 1
    if n % 2 == 0:
        a, b, c, d = sq(a11, a12, a21, a22, MOD)
        return pw(a, b, c, d, n // 2, MOD)
    else:
        a, b, c, d = pw(a11, a12, a21, a22, n - 1, MOD)
        return mu(a, b, c, d, a11, a12, a21, a22, MOD)

MOD = 1000000009
k = 10
a, b, c, d = pw(0, 1, 1, 1, k, MOD)
print(c)
```

Задание такое:

1. Выяснить, что делает функция `f_sl()`.
2. Выяснить, что делает функция `f_fa()`. Запустив в цикле вычисление этих двух функций при `k` от 1 до 30 сделайте предположение о том, одно и то же они вычисляют или нет.
3. Научиться вычислять время, которое требуется на однократное выполнение функции с заданными аргументами. Делается это при помощи функции `perf_counter()` из модуля `time`. Пример вычисления времени работы в приведённом ниже коде.

```
from time import perf_counter

def f(a, b):
    if b == 0:
        return a
    return f(a + 1, b - 1)

# start timer (you can use any variable)
start = perf_counter()

# call the functions to measure
f(456, 987)

# time is the difference between timer after and before call
# print out the formatted time
print(f"time = {perf_counter() - start:.6f}")
```

4. Убедиться, что набор функций в третьем примере вычисляет то же самое для одинаковых `k` и выяснить пределы применимости функции `f_sl`. А именно: при каком значении `k` она начинает работать дольше секунды?
5. Выясните пределы применимости функции `f_fa()` — при каком минимальном `k` она работает дольше секунды?
6. Попробуйте выяснить пределы применимости функции из третьего примера. Например, посчитать `pw(0, 1, 1, 1, k, MOD)` при значениях `k` **существенно** превышающих критические для `f_fa()`.