

Zadanie H

Monitor - Python

Twoim zadaniem jest zaimplementowanie modułu `monitor` służącego do monitorowania wywołań i zwracanych wartości funkcji.

Moduł powinien zawierać:

- Dekorator `monit(level)`,
- Funkcję `report(level, indent = "-->", limits = None)`, która ma wypisać wszystkie dotychczasowe wywołania i zwracane wartości przez funkcje udekorowane `@monit(x)` dla $x \leq level$. Zmienna `limits` jest słownikiem z funkcji w pary liczb. Wtedy należy wypisać tylko te wywołania funkcji których głębokość (liczba niezakończonych wywołań udekorowanych funkcji w tym jednym momencie) mieści się w podanym przedziale, z wcięciem zgodnym z kolejnymi wywołaniami kolejnych monitorowanych funkcji. Formatowanie powinno być dokładnie takie samo jak w teście przykładowym.

Monitorowane funkcje nie będą wywoływane z `keyword arguments`.

- Funkcję `clear()`, która czyści wszystkie zapamiętane wywołania i zwracane wartości.

Wysyłanie rozwiązania

Należy wysłać jeden plik o nazwie **monitor.py**.

Uwagi

Można założyć, że testy nie będą wredniejsze niż test przykładowy :)

Dostępna pamięć: **32MB**

Przykład

Dla testu:

```
from monitor import monit, report, clear
```

```
@monit(1)
def fib(a):
    if a == 0: return 1
    if a == 1: return 1
    return fib(a-1) + fib(a-2)
```

```
fib(6)
report(1, limits = {fib:(1,2)})
report(1,limits = {fib:(6,6)})
clear()
```

```
@monit(2)
def f(a):
    if a == 0:
        return 1;
    return f(a-1)*a
```

```
@monit(1)
def g(a, b):
    if b == 0:
        return 1
    return g(a, b-1) * f(a)
```

$$g(2, 2)$$

```
report(2)
report(1)
report(2, limits = {f:(1,2), g:(1,2)})
clear()
```

Poprawna odpowiedź jest:

```
fib(6)
->fib(5)
->return 8
->fib(4)
->return 5
return 13
->->->->->fib(1)
->->->->->return 1
->->->->->fib(0)
->->->->->return 1
g(2,2)
->g(2,1)
->->g(2,0)
->->return 1
->->f(2)
->->->f(1)
->->->->f(0)
->->->->return 1
->->->return 1
->->return 2
->return 2
->f(2)
->->f(1)
->->->f(0)
->->->->return 1
->->return 1
->return 2
return 4
g(2,2)
->g(2,1)
->->g(2,0)
->->return 1
->return 2
return 4
g(2,2)
->g(2,1)
->return 2
->f(2)
->return 2
return 4
```