

Wstęp do programowania w języku Python

Kolokwium

28 stycznia 2015

Za kolokwium można dostać 104 punkty (100 zapowiedzianych oraz 4 bonusowe). W rozwiązaniach istotna jest przede wszystkim czytelność kodu. Uwaga: we wszystkich zadaniach nie musisz przejmować się polskimi znakami, innymi słowy możesz założyć, że `len("żółć") == 4`

Zadanie 1.(20pkt) Napisz w Pythonie funkcję `prostokat(a,b)`, która za pomocą instrukcji `print` wypisuje prostokąt o szerokości `a` znaków i wysokości `b` znaków. Wykorzystaj tę funkcję do napisania funkcji `schody(N,h,a,krok)`, która wypisuje `N` prostokątnych schodków, o wysokości `h`, z których najmniejszy ma szerokość `a`, a każdy kolejny o `krok` większą. Poniżej wynik działania wywołania funkcji `schody(3,4,6,8)`

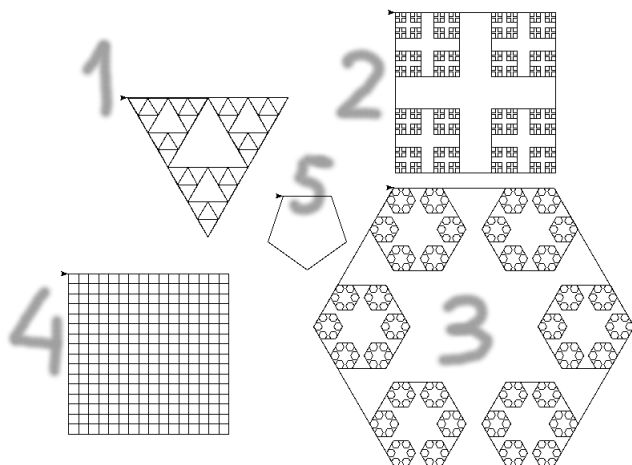
```
*****
*      *
*      *
*****
*****
*          *
*          *
*****
*****
*          *
*          *
*****
*****
```

Zadanie 2.(15pkt) Przeanalizuj poniższe polecenia i opisz, jaką wartość ma lista `L` po zakończeniu działania programu. Jeżeli uważasz, że któreś z poleceń spowoduje błąd, to napisz to i podaj wartość listy `L` przy założeniu, że tego polecenia nie ma.

```
def f(s):
    while len(s) >= 2 and s[0] == s[1]:
        s = s[1:]
    while len(s) >= 2 and s[-1] == s[-2]:
        s = s[:-1]
    return s

L = [0,1,2]
L[0] = f('aaaabsolutna_rewalacja')
L[1] = f('tttttttaakkkkkkk')
L[2] = f('abc')
L.append('abc')
L += 'abc'
```

Zadanie 3.(20pkt) Napisz w Pythonie rekurencyjną funkcję `rysunek`, za pomocą której można wykonywać rysunki takie, jak poniższe. Wybierz takie parametry funkcji, żeby mieć jak największą kontrolę nad wyglądem rysunku. Podaj przybliżone wartości parametrów, które dały takie rysunki.



Zadanie 4.(25pkt) W zadaniu tym będziemy liczyć wystąpienia poszczególnych cyfr w dziesiętnej reprezentacji liczb z pewnego zbioru. Napisz funkcje:

- `policzCyfry(S)`, która dla zbioru `S` zwraca słownik, w którym kluczami są cyfry, a wartościami liczby wystąpień tych cyfr w liczbach ze zbioru `S`.
- `posortujCyfry(S)`, która dla zbioru `S` zwraca napis złożony z 10 cyfr, ułożonych od najrzadziej występującej w liczbach z tego zbioru, do najczęściej występującej. Cyfry występujące tak samo często powinny w napisie występować w kolejności rosnącej.

Przykładowo: `policzCyfry(set(range(22)))` powinno zwrócić

```
{0: 3, 1: 13, 2: 4, 3: 2, 4: 2, 5: 2, 6: 2, 7: 2, 8: 2, 9: 2}
```

natomiast `posortujCyfry(set(range(22)))` powinno zwrócić, "3456789021"

Zadanie 5.(24pkt) W tym zadaniu powinieneś napisać definicje kilku opisanych niżej funkcji. Każdą z nich da się napisać w następującej *wyrazeniowej* postaci:

```
def f(...):
    return wyrażenie-w-pythonie
```

i taka postać jest w tym zadaniu preferowana, tzn. poprawne napisanie definicji zgodnej z tą postacią warte jest 3.5 lub 4p (3.5 w przypadku, gdy rozwiązanie będzie dużo bardziej skomplikowane od rozwiązania modelowego). Poprawna definicja korzystająca z innych konstrukcji (pętle, instrukcje podstawienia, ...) jest warta 3p.

- `sumaKwadratow(N)` – funkcja zwraca sumę kwadratów liczb naturalnych mniejszych od `N`
- `bezPowtorzen(L)` – funkcja zwraca wartość logiczną równą `True` wtedy i tylko wtedy, gdy na liście liczb `L` wszystkie elementy są różne
- `autoanagramowe(S)` – funkcja zwraca zbiór tych słów ze zbioru `S`, które mają następującą własność: mają parzystą długość, a ich lewa i prawa połowa złożone są z takich samych liter (występujących tyle samo razy). Przykładowe słowa autoanagramowe (kreska oznacza połowę): nia-nia, powyła-pywały, katech-etkach.
- `zawieraWielokrotnie(s,k,t)` – funkcja zwraca `True` wtt gdy słowo `s` sklejone `k`-krotnie jest podslowem słowa `t`.
- `posortowana(L)` – zwraca `True` wtt gdy lista `L` jest posortowana
- `posortowana(L)` – zwraca `True` wtt gdy lista `L` jest posortowana. Dodatkowo wymagamy, by czas działania funkcji był liniowy (oczywiście możesz użyć tego samego programu i nie musisz go przepisywać. Wystarczy w odpowiedzi do poprzedniego punktu zaznaczyć, że jest to również odpowiedź do tego podpunktu).