

Kolokwium ze Wstępu do programowania

19 stycznia 2011 r.

Zadanie 1.(15pkt) Napisz w C lub Pythonie funkcję `krzyzyk(n)`, która rysuje krzyżyk z gwiazdek, składający się z 5-ciu kwadratów o boku `n`, jak na poniższym rysunku (na którym pokazany jest wynik wywołania `krzyzyk(4)`).

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

Zadanie 2.(15pkt) Poniżej znajdziesz sekwencję poleceń Pythona. Przeanalizuj tę sekwencję i powiedz, jaką wartość ma lista `L` po zakończeniu działania programu. Uwaga: każde pole listy jest osobnym podzadaniem, jeżeli nie wiesz, jaką wartość ma któreś pole, to wstaw znak `'?'`.

```
>>> def double(s):
...     r = ""
...     for a in s:
...         r += a + a
...     return r

>>> L = "ala ma kota".split()
>>> L[0] = L[0][1:] * 3
>>> L[1] = str(3) + L[1]
>>> L[2] = double(L[2])
>>> L.append([8])
>>> L += [8]
```

Zadanie 3.(15pkt) Niech $a_k = k^2$. Niech $s_n = \sum_{i=0}^n a_i$. Napisz w dowolnym języku fragment kodu, który wczytuje (ze standardowego wejścia) liczbę M i następnie zwraca najmniejszą wartość k , taką że $s_k \geq K$. Nie musisz przejmować się tym, że wykroczysz poza zakres wybranego typu całkowitoliczbowego.

Zadanie 4.(20pkt) Napisz w języku C funkcję `void reverse(char *s)`, która odwraca napis `s`. Czyli jeżeli przed wywołaniem tej funkcji napis `s` miał wartość `abcde`, to po wywołaniu tej funkcji napis `s` powinien mieć wartość `edcba`.

Zadanie 5.(20pkt) Napisz w Pythonie funkcję `addNumbers`, która jako argument bierze listę napisów i zwraca listę o tej samej długości, w której do powtarzających się napisów zostaną dodane kolejne numery, jak na podanych niżej przykładach:

```
>>> addNumbers(["ala", "ma", "kota"])
["ala", "ma", "kota"]
>>> addNumbers(["a", "b", "c", "a", "b", "c", "d", "e"])
["a/1", "b/1", "c/1", "a/2", "b/2", "c/2", "d", "e"]
>>> addNumbers(["pojade", "nad", "morze", "albo", "nad", "jezioro"])
["pojade", "nad/1", "morze", "albo", "nad/2", "jezioro"]
```

Zadanie 6.(15pkt) Rozważmy napisany poniżej program:

```
def glebokoscListy(L):  
    if type(L) != list:  
        return 0  
    if L == []:  
        return 1  
    glebokosci = []  
    for e in L:  
        glebokosci.append(glebokoscListy(e))  
    return 1+max(glebokosci)
```

- a) Jakie wartości zwróci ta funkcja, wywołana dla argumentów: [], [1,2,3], [[1]], [1,[2],[[3]]].
- b) Uzupełnij poniższy ciąg instrukcji w ten sposób, by funkcja glebokoscListy została wywołana, a napis Hurra! się **nie** wyświetlił. W miejscu oznaczonym kropkami możesz wpisywać jedynie różne instrukcje przypisania, nie wolno Ci definiować, ani wywoływać żadnych funkcji.

```
(...)  
glebokoscListy(L)  
print "Hurra!"
```