

## Atsiskaitymas

Atlikti pateiktas uždutis ir įkelti jas Moodle aplinkoje nurodytoje vietoje. Kiekvienos užduoties sprendimas turi būti saugomas atskirame faile, kuris turi būti pavadintas taip: ex{Užduties numeris}.py (1 užduotis – ex1.py, 2 užduotis – ex2.py ir t.t.). Failus į Moodle kelkite **nearchyvuotus**.

Atsiskaitymo metu galima paskaitų medžiaga ir internetiniais ištekliais, tačiau ne kolegų pagalba.

### 1 užduotis (4t.)

Sukurti funkciją **sorted\_and\_multiplied**, kuri priima 2 parametrus: skaičių sąrašą ir daugiklį, kurio numatytoji reikšmė yra 1. Funkcija grąžina sąrašą su didėjimo tvarka surikiuotomis reikšmėmis, kurios yra padaugintos iš nurodyto daugiklio.

Įvestis	Išvestis
<pre>print(sorted_and_multiplied([14, 57, 93, 1], 8)) print(sorted_and_multiplied([8, 23, 4, 11], 2)) print(sorted_and_multiplied([33, 45, 7, 101]))</pre>	<pre>[8, 112, 456, 744] [8, 16, 22, 46] [7, 33, 45, 101]</pre>

### 2 užduotis (3t.)

Sukurti funkciją **pythagorean\_checker**, kuri priima 3 parametrus: dvi stačiojo trikampio kraštines (a, b) ir įžambinę (c). Funkcija grąžina boolean reikšmę ar nurodytos reikšmės atitinka Pitagoro teoremą ( $a^2+b^2=c^2$ ).

Įvestis	Išvestis
<pre>print(pythagorean_checker(3, 4, 5)) print(pythagorean_checker(3, 8, 5)) print(pythagorean_checker(5, 12, 13))</pre>	<pre>True False True</pre>

### 3 užduotis (4t.)

Sukurti funkciją **split\_values**, kuri priima simbolių seką - x, bei vieną skaitmenį - y. Atlikti patikrinimus, jog y tikrai skaičius, jog jis yra didesnis už 0, bei x ilgis dalinasi į lygias dalis po y simbolių. Jei šios sąlygos tenkinamos, funkcija suskaido pateiktą tekstą į lygias dalis po y simbolių ir grąžina sąrašą, kuriame yra pateikiami unikalūs simboliai iš kiekvienos dalies surikiuoti atvirkštine abėcėlės tvarka (nuo Z iki A). Priešingu atveju, funkcija grąžina pranešimą „Error“.

Įvestis	Išvestis
<pre>x = "ABCDEFGH" y = 2 print(split_values(x, y))</pre>	<pre>['BA', 'DC', 'FE', 'HG']</pre>
<pre>x = "AABCADAA" y = 4 print(split_values(x, y))</pre>	<pre>['CBA', 'DA']</pre>
<pre>x = "AABCAAADA" y = "trys" print(split_values(x, y))</pre>	Error
<pre>x = "AABCAAADA" y = 5 print(split_values(x, y))</pre>	Error

### 4 užduotis (2t.)

Sukurti funkciją **pack**, kuri atlieka teksto suspaudimą. Funkcija priima vieną parametą x - tekstas ir grąžina tekstą sudarytą iš simbolio ir jo iš eilės einančių pasikartojimų skaičiaus t.y. suspaudimas vykdomas grupuojant iš eilės einančius simbolius.

Įvestis	Išvestis
<pre>print(pack("aaavvvfdff"))</pre>	<pre>a3v3f1d1f2</pre>
<pre>print(pack("avtvvvff"))</pre>	<pre>a1v1t1v3f2</pre>

## 5 uždutis (2t.)

Sukurti funkciją **number\_operations**, kuri priima vieną parametą a - vienos dimensijos sąrašas. Funkcijoje atliekamas sąrašo filtravimas paliekant reikšmes iš intervalo 10-100. Funkcija grąžina išfiltruotų reikšmių: vidurkį, mažiausią ir didžiausią reikšmes, bei sumą.

Sukurti antrą funkciją **list\_operations**, kuri priima vieną parametą b - sąrašą sudarytą iš sąrašų ir kiekvienam sąrašo elementui iškviečia pirmą funkciją ir grąžina sąrašą sudarytą iš pirmosios funkcijos rezultatų.

Ivestis	Išvestis
<pre>data = [     [1, 10, 34, 110, 400, 30, 20],     [-5, -10, 55, 120, 30],     [2, 67, 23, 78, 200], ] print(*list_operations(data), sep="\n")</pre>	<pre>[23.5, 10, 34, 94] [42.5, 30, 55, 85] [56.0, 23, 78, 168]</pre>
<pre>data = [     [-1, 45, 23, 32, 999],     [67, 99, 23],     [23], ] print(*list_operations(data), sep="\n")</pre>	<pre>[33.333333333333336, 23, 45, 100] [63.0, 23, 99, 189] [23.0, 23, 23, 23]</pre>

## 6 uždutis (3t.)

Sukurti funkciją **get\_acronym**, kuri priima 2 parametrus: tekstą ir jungimo simbolį, kurio numatytoji reikšmė yra yra tuščias string'as ("). Funkcija grąžina tekstą sudarytą iš pirmųjų paduoto teksto didžiųjų raidžių, kuris yra sujungtas nurodytu jungimo simboliu.

Ivestis	Išvestis
<pre>print(get_acronym('Federal Bureau of Investigation')) print(get_acronym('Customs and Border Protection', ' ')) print(get_acronym('Central Intelligence Agency', '.'))</pre>	<pre>FBI C B P C.I.A</pre>

## 7 uždutis (7t.)

Sukurti funkciją **calculate**, kuri priima 3 string tipo parametrus parametrus. Pirmasis paduotas parametras turi būti paverčiamas dvejetainiu skaičiumi, antrasis – dešimtainiu, trečiasis – aštuntainiu. Jei visi trys parametrai gali būti paversti atitinkamais skaičiais, jie yra sudedami ir funkcija grąžina jų sumą šešioliktainiu ir dešimtainiu formatu, jei bent vienas iš parametru negli būti paverstas į skaičių, grąžinamas klaidos pranešimas.

Ivestis	Išvestis
<pre>print(calculate('10', '10', '10')) print(calculate('12', '43', '2')) print(calculate('101', 'a', '11')) print(calculate('1', '20', 'z')) print(calculate('10', '85', '11'))</pre>	<pre>The result is 0x14 in hex or 20 in dec 12 is not a valid binary integer a is not a valid decimal integer z is not a valid octal integer The result is 0x60 in hex or 96 in dec</pre>