

NTI/PIPA - Programovací jazyk Python (2023)

Dashboard / Courses / FM / NTI / 2023/24 / NTI/PIPA - Programovací jazyk Python (2023) / Úkoly a cvičení / Úkol 3. - transformace dat

My courses

- ITE/CITE - Číselná technika (2022)
- ITE/EDK - Elektronická dokumentace (2022)
- ITE/MTLB - Výpočty, simulace a vizualizace Matlab (2022)
- ITE/SGI - Signály a informace (2023)
- ITE/ZKO - Základy konstruování (2023)
- KAP/AJLA - Úvod do lin. algebry a diskrétní mat. (2022)
- MTI/AUG1 - Algoritmizace a programování 1 (2022)
- Samostatné úlohy z předmětu Algoritmizace a programování 1 (2022)
- MTI/AUG2 - Algoritmizace a programování 2 (2022)
- MTI/ICP - Číselové počítče (2023)
- MTI/IOS - Databázové systémy (2023)
- MTI/JPJC - Programování v jazyce C/C++ (2023)
- MTI/JSIN - Softwarové inženýrství (2023)
- MTI/UDJ - Úvod do inženýrství (2022)
- MTI/NAWP - Vývoj aplikací pro Windows (2023)
- MTI/VALD - Algoritmizace a datové struktury (2023)
- MTI/OPS - Operační systémy (2023)
- NTI/PIPA - Programovací jazyk Python (2023)
- Počítačové sítě
- NTI/PST - Počítačové sítě (2022)
- ITE/ISH - Úvod do Shellu (2022)
- MTI/TWS - Tržba WWW stránek (2023)
- NTI/USA - Úvod do statistické analýzy (2023)

Navigation

- Dashboard
 - Site home
 - Site pages
 - Courses enrollment (STAG)
 - Courses unenrollment
 - Propojení se STAGem
- My courses
 - ITE/CITE - Číselná technika (2022)
 - ITE/EDK - Elektronická dokumentace (2022)
 - ITE/MTLB - Výpočty, simulace a vizualizace Matlab ...
 - ITE/SGI - Signály a informace (2023)
 - ITE/ZKO - Základy konstruování (2023)
 - KAP/AJLA - Úvod do lin. algebry a diskrétní mat. (2...
 - MTI/AUG1 - Algoritmizace a programování 1 (2022)
 - Samostatné úlohy z předmětu Algoritmizace a progra...
 - MTI/AUG2 - Algoritmizace a programování 2 (2022)
 - MTI/ICP - Číselové počítče (2023)
 - More...
- Courses
 - FM
 - DFM
 - ITE
 - MTI
 - NTI
 - 2023/24
 - NTI/ADA - Algoritmy a datové struktury (2023)
 - NTI/ATIP - Automaty a formální jazyky (2023)
 - NTI/VALD - Algoritmizace a datové struktury (2023)
 - NTI/AMP - Alternativní metody programování (2023)
 - NTI/ARMO - Aplikace počítačových modelů (2023)
 - NTI/ARP - Architektura počítačů (2023)
 - NTI/CFD - Výpočetní mechanika tekutin (2023)
 - NTI/DPG - Distribuované programování (2023)
 - NTI/EMM - Experimentální metody v mechanice (2023)
 - NTI/JPD - Jazyky pro popis dat (2023)
 - NTI/KAS - Kybernetická bezpečnost a šifrování (2023)
 - NTI/PIPA - Programovací jazyk Python (2023)
 - Participants
 - Competencies
 - Grades
 - Programovací jazyk Python - PIPA LS 2024
 - Úkoly a cvičení
 - Úkol 0 - přihlaste se na github/tul.cz
 - Úkol 1 - první program
 - Úkol 2 - konvenční čtyřúhelník
 - Úkol 3. - transformace dat
 - Úkol 4. - Caesarova šifra
 - Úkol 5. - algoritmizace problému
 - Úkol 6. - regulární výrazy
 - Úkol 7. - zpracování JSON a HTML dat
 - Úkol 8. - Poker (starší zkoušková otázka)
 - Úkol 9. - Binární Vyhledávací Strom
 - Úkol 10. - Cenzor (starší zkoušková otázka)
 - Účast na přednáškách
 - Ponoříme se do Pythonu - úvod do předmětu
 - proměnné a konstanty
 - 3. strukturované datové typy - kolekce a sekvence
 - 4. další vlastnosti jazyka
 - 5. testování kódu
 - 6. standardní textové formáty a jejich zpracování
 - 7. funkce a jejich pokročilé využití
 - 8. tvorba vlastních typů, principy OOP
 - 9. tvorba aplikací s CLI (command line interface)
 - 10. další moduly standardní knihovny jazyka Python
 - 11. Vykonnost Python programů
 - Topic 14
 - 2022/23
 - 2021/22
 - 2020/21
 - 2019/20
 - 2018/19
 - Aplikace GIS
 - Diplomové a bakalářské práce 2021/22
 - Geografické informační systémy
 - Kopetschke DP, BP, PRO, PRJ 2016/17
 - Počítačové sítě
 - RSS
 - Kurzy mimo STAG
 - Bezpečnost práce na elektrickém zařízení v laborat...
 - Admission Test Mechatronics 2024
 - Samostatný elektrotechnik pro elektromagnetickou k...
 - Násklání geometrie
 - Hodnocení kvality výuky (BS-IT 2021/22)
 - Virtuální setkání akademické obce FM
 - Připrava na přijímačky z informatiky
 - Připrava na přijímačky z matematiky
 - Studentská konference Fakulty mechatroniky
 - MTI/CSHARP - TI
 - Podnikový informační systém SAP
 - NÁVODY, MANUÁLY
 - Další podpůrné materiály
 - Kurzy pro zaměstnance TUL
 - Externí kurzy
 - FA
 - FE
 - FP

Úkol 3. - transformace dat

Opened: Sunday, 2 October 2023, 11:59 PM
Due: Tuesday, 19 March 2024, 10:59 PM

Při řešení tohoto úkolu si prověřte práci se několika datovými typy. Můžete si také vyzkoušet vytváření seznamů a slovníků pomocí generatorové notace (list / dict comprehension)

Cílem úkolu je vytvořit program, který transformuje data z jedné struktury do druhé.

Na vstupu jsou dány 3 sekvence (line_a, line_b, line_c). Každá z nich obsahuje několik uspořádaných dvojic uložených jako tuple (id, count). Sekvence mohou mít různou délku.

Sekvence může tedy vypadat například takto: ((1, 3), (7, 5), (11, 3), (16, 2)). První prvek sekvence je tedy tuple s hodnotami id = 1, count = 3.

Oba prvky (id i count) jsou vždy celá čísla (int). Hodnoty count se mohou opakovat, **id je ale unikátní a v každé sekvenci se vyskytuje nejvýše jednou**.

Vášim úkolem je **spojit tyto tři sekvence** do jednoho slovníku. Ten bude výstupem z programu.

Položky slovníku budou seznamy hodnot count, klíče pak id. A to v následujícím tvaru: {id: [A, B, C]}, kde A, B a C jsou hodnoty pro příslušné id v první, druhé a třetí sekvenci. Toto pořadí je potřeba zachovat. První pozice v seznamu tedy vždy patří první sekvenci, druhá pozice druhé atd.

Ověřte pozor - neptalí, že každé id je obsaženo ve všech sekvencích. Může být ve všech, ve dvou, nebo pouze v jedné. Pokud id v některé sekvenci není, ve výsledku bude pod daným klíčem v seznamu na místě pro příslušnou sekvenci 0. A samozřejmě, pokud id není v žádné ze sekvencí, ve výsledku nebude také.

Pro lepší názornost se podívejme na jeden konkrétní příklad:

Zadané sekvence mají následující podobu:

```
line_a = [(1, 3), (5, 4), (10, 2)]
line_b = [(1, 2), (2, 4), (6, 2), (12, 3)]
line_c = [(1, 5), (3, 2), (7, 3)]
```

První a třetí sekvence obsahují tři dvojice, druhá sekvence čtyři. Id 1 je ve všech třech sekvencích, id 2 a 5 ale jen ve druhé (line_b). Dále například id 7 je pouze ve třetí z nich a tak dále.

Transformací sekvencí musí vzniknout následující slovník:

```
{1: [1, 2, 5],
 2: [0, 4, 0],
 3: [4, 0, 2],
 5: [0, 2, 0],
 7: [0, 0, 3],
10: [2, 0, 0],
12: [0, 3, 0]}
```

Poznamka: id jsou zde pro lepší názornost seřazena, ale výsledek vaší funkce uspořádaný být nemusí. Klíče v normálním slovníku jsou neuspořádané.

Zopakujme ještě jednou, že každá sekvence může obsahovat různá id. Pokud se pro určité id hodnota v některé sekvenci nevyskytuje, použijte se nula (výchozí hodnota). Ještě jednou zopakujeme, že sekvence nemusí mít stejnou délku, jak je vidět v příkladu.

Stejně jako v minulém případě najdete ve vzorovém repozitáři šablonu pro řešení a test pro usnadnění vývoje.

Řešení musí splňovat tyto podmínky:

1. **Musí jít o testovatelný kód** - vše tedy musí být ve funkcích, nebo uzavřeno v podmínce `__name__ == "__main__"`
2. Modul musí být v adresáři `cv03` a musí se jmenovat `merger.py`. Funkce pro spojení sekvencí se musí jmenovat `merge_tuples` a musí mít právě tři parametry. Tady samozřejmě opět doporučuji využít šablonu z repozitáře zdrojových kódů.
3. Pro řešení vám musí stačit standardní instalace Pythonu (bez dalších třetích stran)
4. Výsledný kód musí při testu programem PyLint se standardním nastavením získat alespoň 8 bodů. Práci s programem PyLint jsme si ukázali na přednášce, případné dotazy samozřejmě rád zodpovím.
5. Kód jako obvykle odezdejte na github/tul.cz do svého repozitáře.

Submission status

Submission status	This assignment does not require you to submit anything online
Grading status	Graded
Time remaining	The due date for this assignment has now passed
Last modified	-
Submission comments	Comments (0)

Feedback

Grade	5.00 / 5.00
Graded on	Wednesday, 27 March 2024, 1:51 PM
Graded by	LM Lukáš Mádl
Feedback comments	V pořádku

← Úkol 2. - konvenční čtyřúhelník

Jump to...

Úkol 4. - Caesarova šifra →

- > FS
- > FT
- > FZS
- > Rektorát
- > UKN
- > UZS
- > Velejné kurzy
- > _ARCHIV

You are logged in as Martin Šimon (Log out)
NTU/PIPA (2023)
[Get the mobile app](#)