

NTI/PIPA - Programovací jazyk Python (2023)

Dashboard / Courses / FM / NTI / 2023/24 / NTI/PIPA - Programovací jazyk Python (2023) / Úkoly a cvičení / Úkol 4 - Caesarova šifra

My courses

- ITE/CITE - Číslicová 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/JULA - Úvod do lin. algebry a diskrétní mat. (2022)
- MTI/ALG1 - Algoritmizace a programování 1 (2022)
- Samostatné úlohy z předmětu Algoritmizace a programování 1
- MTI/ALG2 - Algoritmizace a programování 2 (2022)
- MTI/ICP - Číslicové počítače (2023)
- MTI/IGS - Databázové systémy (2023)
- MTI/PRJ - Programování v jazyce C/C++ (2023)
- MTI/STIN - Softwareové inženýrství (2023)
- MTI/UDI - Úvod do inženýrství (2022)
- MTI/NAPW - Vývoj aplikací pro Windows (2023)
- NTI/ALD - Algoritmizace a datové struktury (2023)
- NTI/OPS - Operační systémy (2023)
- NTI/PIPA - Programovací jazyk Python (2023)
- Počítačové sítě
- NTI/PST - Počítačové sítě (2022)
- NTI/SH - Úvod do Shellu (2022)
- NTI/TWS - Tvorba WWW stránek (2023)
- NTI/USA - Úvod do statistické analýzy (2023)

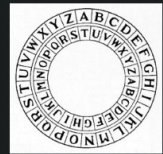
Navigation

- Dashboard
 - Site home
 - Site pages
 - Courses enrollment (STAG)
 - Courses unenrollment
 - Propojení se STAGem
 - My courses
 - ITE/CITE - Číslicová 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/JULA - Úvod do lin. algebry a diskrétní mat. (2...
 - MTI/ALG1 - Algoritmizace a programování 1 (2022)
 - Samostatné úlohy z předmětu Algoritmizace a progr...
 - MTI/ALG2 - Algoritmizace a programování 2 (2022)
 - MTI/ICP - Číslicové počítače (2023)
 - More...
 - Courses
 - FM
 - DFM
 - ITE
 - MTI
 - NTI
 - 2023/24
 - NTI/ADA - Algoritmy a datové struktury (2023)
 - NTI/ATP - Automaty a formální jazyky (2023)
 - NTI/ALD - Algoritmizace a datové struktury (2023)
 - NTI/AMP - Alternativní metody programování (2023)
 - NTI/ARMO - Aplikace počítačových modelů (2023)
 - NTI/IARP - 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 gitaabul.cz
 - Úkol 1 - první program
 - Úkol 2 - kmenování čtyřlístek
 - 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. Výkonnost 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...
 - Měření geometrie
 - Hodnocení kvality výuky (BS-IT 2021/22)
 - Virtuální setkání akademické obce FM
 - Připrava na přijímací zkoušky z informatiky
 - Připrava na přijímací zkoušky 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 4. - Caesarova šifra

Opened: Sunday, 2 October 2023, 12:00 AM
Due: Tuesday, 26 March 2024, 10:55 PM

Caesarova šifra je klasickým příkladem tzv. substituční šifry. V porovnání s moderními šiframi je její princip velice jednoduchý. Vstupem je řetězec, který má být zašifrován a dále číslo, udávající znakový posun (offset).



Takže například je-li vstupem slovo **ahoj** a offset = 1 pak výstupem je slovo **biqk**. Každý znak se posune o jedničku - z **a** bude **b** a tak dále.

Při offsetu 13 je výsledkem slovo **nubw**. Pozor je potřeba si dát na konec abecedy - z se při offsetu 1 změní na a, při offsetu 2 na b atd. Více si o principu i historii této šifry přečtete například na Wikipedii

Cvičení je již za 10 bodů a má dvě části.

Část 1.

Vášim úkolem je nejprve vytvořit dvě funkce - encrypt a decrypt. Funkce encrypt na vstupu přijme dva parametry - řetězec a offset a jako výstup vrátí nový řetězec. Obdobně funkce decrypt přijme řetězec a offset a vrátí řetězec původní.

Pro zjednodušení postačí, když funkce bude šifrovat pouze písmena - z ASCII tabulky. Nemusíte tedy řešit české, ani jiné diakritické znaky. Funkce ovšem musí korektně zpracovat **velká a malá písmena** - tedy zachovat jejich velikost i v šifrovaném textu.

Pozor na to, že zadaný **offset může být také větší než 26** - což je počet písmen bez diakritiky. Ostatní znaky, jako například mezery nebo interpunkci, by funkce měla zachovat.

Funkce crypt a decrypt jsou navzájem inverzní - což můžete využít při testu aplikace. K testu můžete využít také to, že nejčastější varianta offsetu - 13 je jedním z podporovaných kódování balíčku `codecs`. Zkuste si například `codecs.encode('ahoj', 'rot13')`.

Příklad řešení:

vstup = 'Ahoj blizne cos cekal?'

offset = 13

encrypt(vstup) == 'Nubw oymnar pbf prmy?'

Část 2.

Funkci encrypt následně použijte k **zašifrování souboru source.txt**, který najdete v šabloně cvičení v repozitáři.

Program musí po spuštění **soubor načíst, vypočítat z dat offset a následně uložit soubor result.txt**, se zašifrovaným textem.

Jména obou souborů - source.txt a result.txt zadáte v programu přímo jako konstanty.

Jak vypočítat offset? Implementujte následující postup:

- spočítejte počet slov pro jednotlivé řádky. *Poznámka: protože nás nezajímá délka jednotlivých slov, nemusíte při dělení řádku na jednotlivá slova řešit interpunkční znaménka a další speciální znaky.*
- spočítejte průměrný počet slov pro všechny řádky souboru
- výsledek zaokrouhlete dolů na celé číslo a máte hledaný offset.

Pro kontrolu uveďte, že **pro vzorový soubor je tato hodnota 11**. Při hodnocení bude použitý jiný obsah souboru, proto musí program tuto hodnotu vypočítat dynamicky. Nebude ale obsahovat žádné jiné znaky, než ty které najdete ve vzorovém souboru.

Konkrétní implementace druhé části je ve vaší režii a další funkce či metody, které vytvoříte, si můžete pojmenovat jak potřebujete.

Výsledné řešení splňovat také následující podmínky:

- musí jít o testovatelný kód - vše tedy musí být buď ve funkcích, nebo uzavřeno pod `__name__ == "__main__"`
- modul musí být v adresáři `04a` a musí se jmenovat `caesar.py`. Hodnocené funkce se musí jmenovat `encrypt` a `decrypt`. Opět můžete využít šablonu ze vzorového repozitáře.
- pro řešení vám musí stačit standardní instalace pythonu (bez balíčků třetích stran)
- výsledný kód musí při testu programem PyLint se standardním nastavením získat alespoň 8 bodů. Tento úkol je poslední u kterého ještě nebudou strhávány body. Pokud dosáhnete horšího hodnocení, dostanete pouze upozornění. Od cvičení 5 už ale bude za každý bod pod 8 minus jeden bod z celkového hodnocení úkolu.

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	10.00 / 10.00
Graded on	Wednesday, 27 March 2024, 3:27 PM
Graded by	LM Lukáš Mázi
Feedback comments	V pořádku

→ Úkol 3. - transformace dat

Jump to...

Úkol 5. - algoritmizace problému →

- > 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](#)