

תרגיל בית 4

1. (50%) בשאלה זאת תממשו צופן לטקסט הכתוב באותיות לטיניות

א. נגדיר פעולת חיבור (שנסמן אותה +) על האותיות 'z', 'y', 'x', 'w', 'v', 'u', 't', 's', 'r', 'q', 'p', 'o', 'n', 'm', 'l', 'k', 'j', 'i', 'h', 'g', 'f', 'e', 'd', 'c', 'b', 'a' לכל אות מותאם ערך מספרי: 'a' מקבלת ערך 0, 'b' מקבלת את הערך 1, וכן הלאה עד 'z' שמקבלת ערך 25. כדי לחבר שתי אותיות מחברים את הערכים המספריים שלהן ואת התוצאה מתרגמים לאות המתאימה. אם התוצאה גדולה מ 25 לוקחים את השארית מ 26. למשל, כדי לחבר 'f' ו 'x' מחשבים $23+5=28$, השארית מ 26 היא 2 ולכן התוצאה היא האות 'c'. דוגמאות נוספות: 'e'='g'+ 'y', 'd'='c'+ 'b', 'a'='a'+ 'a'. כתבו פונקציה `add_letters` במקבלת שתי מחרוזות. אם שתי המחרוזות באורך 1 ומייצגות אותיות לטיניות הפונקציה תחזיר את הסכום של האותיות לפי ההגדרה הנ"ל. אם לא – יוחזר `None`. התוצאה תמיד תהיה אות לטינית קטנה, גם אם אחד הפרמטרים או שניהם מייצגים אותיות לטיניות גדולות. דוגמאות:

```
add_letters('a', 'b') # 'b'
add_letters('x', 'f') # 'c'
add_letters('y', 'g') # 'e'
add_letters('a', 'bcd') # None
add_letters('%', ' ') # None
```

ב. כתבו פונקציה `add_strings` שמקבלת שתי מחרוזות המורכבות מאותיות לטיניות באזכורים כלשהם ומחזירה מחרוזת שבה כל תו הוא סכום התווים המתאימים במחרוזות הקלט. אורך הפלט כאורך המחרוזת הקצרה מבין השתיים. למשל `add_strings('input', 'output')` תחזיר `whijn` כי `'h'='u'+ 'n'`, `'w'='o'+ 'i'` וכן הלאה. אם מחרוזות הקלט לא מורכבות אך ורק מאותיות לטיניות הפונקציה תחזיר `None`. יש להשתמש בפונקציה `add_letters` מסעיף א'.

ג. [צופן ויז'נר](#) (Vigenere) הוא צופן עתיק מהמאה ה 16 להצפנת טקסט הכתוב באותיות לטיניות. כדי להצפין טקסט כלשהו `s` בוחרים מפתח סודי `k` (בדרך כלל מלה שקל לזכור), בונים מחרוזת `t` ע"י שיכפול של `k` מספר פעמים עד שהאורך של `t` הוא לפחות באורך של `s` ואז מחברים את `s` ו `t` באמצעות הפונקציה `add_strings`. התוצאה היא הטקסט המוצפן. למשל, אם אנחנו רוצים להצפין את הטקסט `'attackatsixoclock'` והמפתח `k='input'`, אז המחרוזת `t` תהיה `'inputinputinputinput'` ותוצאת ההצפנה תהיה `'igiuvsnimbfbrfhkx'` יש לכתוב פונקציה `vigenere_encrypt` שמקבלת מחרוזת `s` ומפתח `k` ומחזירה את תוצאת ההצפנה של `s` באמצעות המפתח `k`. אם `k` לא מורכב מאותיות לטיניות בלבד, אז יוחזר `None`. אם `s` לא מורכב מאותיות לטיניות בלבד, אז הפונקציה תתעלם מכל התווים שאינם אותיות לטיניות (הפונקציה תתעלם גם מרווחים). הפונקציה `vigenere_encrypt` תשתמש בפונקציה `add_strings`.

למשל, `vigenere_encrypt('Attack at Six (6) o'clock', 'input')` תצפין את המחרוזת `'AttackatSixoclock'` ותחזיר את המחרוזת `'igiuvsnimbfbrfhkx'`. (שימו לב שהפונקציה `add_letters` הופכת אותיות גדולות לקטנות) יש לכתוב פונקציה `vigenere_decrypt` שמקבלת מחרוזת `w` ומפתח `k` ומחזירה את תוצאת הפיענוח של `w` באמצעות המפתח `k`. אם `k` לא מורכב מאותיות לטיניות בלבד, אז יוחזר `None`.

למשל, `vigenere_decrypt('igiuvsnimbfbrfhkx', 'input')` תחזיר את המחרוזת `'attackatsixoclock'` (שימו לב שהתוצאה תמיד באותיות קטנות ובלי רווחים). יש לכתוב תוכנית `vigenere.py` שעושה את הפעולות הבאות, לאחר שהמשתמש התבקש לבחור בין `e` ל `d`:

i. אם המשתמש הקיש `e` (קיצור של `encrypt`), אז התוכנית תבקש ממנו מפתח הצפנה ושם קובץ. התוכנית תצפין באמצעות הפונקציה `vigenere_encrypt` את תוכן הקובץ. התוצאה

- תודפס לקובץ בשם דומה לקובץ המקורי, אבל עם סיומת vig. למשל, אם שם הקובץ המקורי הוא pooh.txt אז תוצאת ההצפנה תישמר לקובץ pooh.vig. התוכנית תצפין רק אותיות אנגליות ותתעלם מכל תו שאינו אות אנגלית.
- ii. אם המשתמש הקיש d (קיצור של decrypt) אז התוכנית תבקש ממנו מפתח הצפנה ושם קובץ ותפענח את הטקסט שנמצא בתוך הקובץ באמצעות הפונקציה vigenere_decrypt. תוצאת הפיענוח תודפס למסך.
- iii. אם המשתמש הקיש כל דבר אחר, אז התוכנית לא תעשה כלום ותפסיק את פעולתה
- התוכנית תתריע במקרה שקובץ כלשהו שהיא אמורה לקרוא ממנו לא נמצא. כדי שהתוכנית לא תיפול במקרה שקובץ לא נמצא יש להשתמש בטיפול בחריגות:

```
try:  
    f = open(filename,'r')  
    s = f.read()  
    f.close()  
except IOError:  
    print("File not accessible")
```

לצורך בדיקה עצמית מצורפים כמה קבצים מוצפנים. המפתחות לפיענוח הקבצים הם:

קובץ	מפתח
alibaba.vig	robbers
catcher.vig	childhood
hamlet.vig	tragedy
poohbear.vig	character

נסו לפענח את הקבצים באמצעות התוכנית שאתם כותבים. ודאו שאתם מקבלים טקסט באנגלית (באותיות קטנות וללא רווחים)

2. (50%) יש לכתוב תוכנית grades.py שקוראת נתונים משני קבצים:

- קובץ students.txt שבו מופיעים מספרי ת"ז ושמות סטודנטים. לדוגמה:

121212121 John Lennon
115116117 Yoko Ono
113113113 Madonna
123123123 Ringo Starr

- קובץ grades.txt שבו בכל שורה מופיע מספר ת"ז ורשימת ציונים (מספרים בין 0 ל 100). למשל,

113113113 80 95 67 100 56
123123123 78 65 43
115116117 77 54 69 89 95 98
121212121 99 55 77 95

התוכנית תקרא את הנתונים לשני מילונים:

- (i) מילון שבו המפתחות הם מספרי ת"ז והערכים הם שמות
- (ii) מילון שבו המפתחות הם מספרי ת"ז והערכים הם רשימות של ציונים.

התוכנית תדפיס למסך את הנתונים הבאים:

- i. שמות הסטודנטים, כל סטודנט בשורה נפרדת, וליד כל שם של סטודנט - ממוצע הציונים שלו. הסטודנטים יופיעו ממיינים לפי ממוצע הציונים של כל סטודנט מהגבוה לנמוך.
- ii. הציון (או הציונים) שהתקבל(ו) הכי הרבה פעמים
- iii. רשימת הציונים שהתקבלו אצל יותר מסטודנט אחד. (לצורך מימוש הסעיף הזה יש לכתוב פונקציה get_common_elements שמקבלת רשימה של רשימות ומחזירה רשימה של האברים שהופיעו ביותר מרשימה אחת. יש להשתמש בפעולות על קבוצות intersection | union)

מספר ת"ז הוא מחרוזת של 9 ספרות.

התוכנית תתריע ותפסיק את פעולתה במקרים הבאים:

- 1) אם יש מס' ת"ז שאינו תקין (לא מורכב מ 9 ספרות)
- 2) אם מס' ת"ז כלשהו מופיע בקובץ אחד ובשני לא
- 3) מס' ת"ז שלא מופיעים לידו ציונים בקובץ grades.txt או מספר ת"ז שאין לידו שם בקובץ students.txt

לצורך פתרון השאלה עליכם לכתוב את הפונקציה get_common_elements שמקבלת רשימה של רשימות ומחזירה רשימה של האברים שהופיעו ביותר מרשימה אחת. יש להשתמש בפעולות על קבוצות intersection | union.

דוגמה: עבור הקבצים students.txt | grades.txt המצורפים הפלט יהיה:

Dani Sanderson 84.75
Yoko Ono 80.33333333333333
Madonna 79.6
John Lennon 73.875
Ringo Starr 70.25
Julianne Moore 68.5

Jennifer Lawrence 61.0
Bob Dilan 56.75
matthew mcconaughey 56.25
Don Quixote 56.0
Most common grades: [95, 67]
Grades that more than one student got:
98, 67, 99, 77, 15, 54, 57, 91, 62, 95

הנחיות הגשה :

- 1- יש להגיש תוכניות שרצות ללא שגיאות. תוכנית שתוגש עם שגיאות תקבל לכל היותר חצי מהנקודות.
- 2- יש לכתוב הערות לתוכנית : docstring בתחילת כל פונקציה, הסבר קצר בתחילת התוכנית, הסבר בתחילת לולאות.
- 3- אין להשתמש במודולים מלבד מודולים סטנדרטיים כמו math, random, sys.
- 4- יש לפתור כל שאלה בקובץ נפרד עם סיומת .py.
- 5- יש להגיש את כל הקבצים בקובץ אחד מכוון עם סיומת zip. שם קובץ ה zip צריך להיות מספר הת"ז שלכם ומספר עבודת הבית. למשל, 22222222_hw4.zip
- 6- כל קובץ יתחיל בהערה ובה המידע הבא :
 - א. שם הסטודנט
 - ב. מס' תעודת זהות
 - ג. מספר דף התרגילים
 - ד. שם התוכניתלמשל, עבור תרגיל 2 בדף 4 :

```
""""  
Student: BoB Marley  
ID: 121212121  
Assignment no. 4  
Program: grades.py  
""""
```

שימו לב : יש להקפיד על הנחיות ההגשה האלה. הגשה שלא בדיוק בפורמט הזה לא תקבל את מלוא הנקודות ואף עלולה להיפסל.