אין לשנות את שם הפונקציות או את החתימה שלהן.

עבור כל אחד מהפתרונות (למעט שאלה 5 ) יש לכתוב unittest (תוך שימוש ב- import unittest) עם שתיים שלוש דוגמאות לפחות. לטסטים יינתן משקל בציון.

בכל השאלות מותר להניח שהארגומנטים הם תקינים.

שאלה 1

כתבו פונקציה (matrix,k=1) half.

**גוף הפונקציה צריך לכלול שורה בודדת** (list comprehension).

הפרמטר matrix זו מטריצה. המטריצה מיוצגת ע"י רשימה (list) של רשימות כאשר כל רשימה פנימית מייצגת שורה אחת של המטריצה. מספר השורות לא בהכרח שווה למספר העמודות של המטריצה.

הפונקציה מחזירה את matrix עם השינויים הבאים עבור k=1: השורה הראשונה נשאר רק האיבר הראשון. מהשורה השנייה נשארו שני איברים ראשונים . מהשורה השלישית נשארו שלושה איברים וכן הלאה.

לדוגמא אם K=1 והמטריצה נתונה :

[[1, 2, 3, 4, 5],

[6, 7, 8, 9, "spam"],

[11, 12, 13, 14, 15],

[16, "stam", 18, 19, 20]]

אז הפונקציה תחזיר:

[[1 ],

[6, 7, ],

[11, 12, 13 ],

[16, "stam", 18, 19 ]]

עבור k=0 תנקה את החלק התחתון של המטריצה:

[[1, 2, 3, 4, 5],

[ 7, 8, 9, "spam"],

[ 13, 14, 15],

[ 19, 20]]

אם matrix היא הרשימה הריקה אז יוחזר [].

שאלה 2

כתבו פונקציה עם החתימה decrypt (string, key=1) שמקבלת מחרוזת (המכילה אותיות קטנות בא"ב האנגלי) ו-"מפתח הצפנה" כארגומנטים ומחזירה את המחרוזת . בשיטת הצפנה זו כל אות מוחלפת באות אחרת – זו שמופיעה בסדר הרגיל של אותיות הא"ב האנגלי k מקומות אחרי האות המוחלפת. k הוא מפתח ההצפנה. למשל אם k הוא 1 אז האות a תוחלף ב- b, b תוחלף ב- c וכן הלאה. (האות z תוחלף ב- a). decrypt("tqbn") (הפעולה ההפוכה להצפנה) אמורה להחזיר "spam". דוגמא נוספת: אם מפתח ההצפנה הוא 3 אז האות a תוחלף ב- d, האות b תוחלף ב- e וכן הלאה. (האות w תוחלף ב- z, x תוחלף ב- a, y תוחלף ב- b וכן הלאה). במקרה זה decrypt("vsdp ", 3) (הפעולה ההפוכה להסבר ההצפנה) אמורה להחזיר "spam".

**הדרכה**:

ניתן להשתמש בפונקציות chr ן- ord כדי להמיר תו (מחרוזת עם תו אחד – אין ב- python טיפוס char) למספר שלם ולהיפך. (הן מוסברות ב- <https://docs.python.org/3/library/functions.html>).

למשל chr(ord('0') + 3) תחזיר את המחרוזת '3'.

ברירת המחדל של הפונקציה תהיה 3 ומחרוזת מוצפנת שתוצאתה היא . “solongandthanksforthefish”,

שאלה 3

כתבו generator function בשם merge שמקבל שני iterables כארגומנטים.

merge מייצרת סדרה ממוינת שמתקבלת ממיזוג הסדרות שמניבים שני ה- iterables.

יש להניח שהסדרות ממוינות ושניתן להשוות בין איברים הלקוחים משתי הסדרות בעזרת האופרטור

<=. שימו לב שהסדרות עשויות להיות באורך שונה.

דוגמא :

def divisable\_by (n, limit):

k = 0

while k < limit:

yield k

k += n

>>> g = divisable\_by(4, 21)

>>> for i in merge(g, [2, 3, 7, 10, 11]):

print(i, sep=' ')

0 2 3 4 7 8 10 11 12 16 20

בדוגמא זו אחד ה- iterables שמועבר כארגומנט הוא בעצמו תוצר של generator function

(הפונקציה divisable\_by) אבל זה לא משנה ל- merge.

הנחיה: כנראה שבמקרה זה לא ניתן להסתפק באיטרציה אוטומטית של ה- irterables (למשל ע"י

לולאת for ) ויש צורך לקרוא ל- iter ול- next בצורה מפורשת.

שימו לב שלפונקציה next יש ארגומנט שני אופציונלי שנוח להשתמש בו אם לא רוצים לטפל

ב- StopIteration -- ראו תעוד בקישור https://docs.python.org/3/library/functions.html#next

שאלה 4

כתבו פונקציה rank(file\_name, how\_to\_rank='total').

הארגומנט הראשון הוא שם של קובץ טקסט המכיל פרוט של המדינות שזכו במדליות באולימפיאדות.

הפונקציה מחזירה אוביקט שאיטרציה עליו תניב את שמות המדינות הזוכות במדליות ממוינות

לפי הניקוד שקיבלו. שיטת הניקוד מפורטת ע"י הארגומנט how\_to\_rank. סדר המדינות באיטרציה יהיה מניקוד גבוה לנמוך. הסדר בין מדינות עם ניקוד זהה הוא שרירותי.

אם how\_to\_rank הוא 'total' אז המספר הכולל של מדליות בהן זכתה המדינה יהיה הניקוד שלה.

למשל מדינה שזכתה בארבע מדליות זהב, חמש מדליות כסף ועשר מדליות ארד תקבל ניקוד

4 + 5 + 10 = 19 ,

אם how\_to\_rank הוא 'weighted' אז לכל מדינה ינתן ניקוד משוקלל: מדלית זהב שווה 3 נקודות, מדלית כסף שווה 2 נקודות ומדלית ארד שווה נקודה אחת. המדינה מהדוגמא הקודמת תקבל הפעם ניקוד 4 \* 3 + 5 \* 2 + 10 \* 1 = 32

אם how\_to\_rank הוא 'gold' אזי סדר המדינות יהיה לפי כמות הזהב והניקוד בהתאם .

כל שורה בקובץ מכילה שם של מדינה ושלושה מספרים המציינים את מספר המדליות מכל סוג בהן זכתה המדינה. רווחים מפרידים בין השדות בשורה. למען הפשטות נניח שאין רווחים בשם המדינה

למשל בריטניה תרשם כ- Great-Britain (עם מקף) במקום Great Britain (עם רווח באמצע).

דוגמא לקובץ קלט (ארה"ב למשל זכתה ב-1022 מדליות זהב, 794 כסף ו- 705 ארד):

Israel 1 1 7

USA 1022 794 705

Great-Britain 263 295 289

הנה דוגמא לשימוש בפונקציה (נניח שלקובץ הקלט בדוגמא הנ"ל קוראים ('winners.txt':

>>> for nation in rank('winners.txt', 'total'):

print(nation)

הפלט יהיה:

"USA":2521

"Great-Britain":847

"Israel":9

אם בדוגמא זו נחליף את הקריאה rank('winners.txt', 'total'): בקריאה

rank('winners.txt', 'weighted')

אז הפלט לא ישתנה עם ניקוד שונה .

שאלה 5 GUI

כתבו ממשק GUI אחד עבור כל השאלות הקודמות (יכול להיות גם מעוצב כתיקייה) , אין שום מגבלה על

העיצוב, או ספריית ה GUI, ובתנאי שעומד בדרישות כדלהלן:

דרך הממשק ניתן יהיה לראות פר שאלה:

1. את השאלה (באנגלית ובקצרה).
2. בבקר נפרד הוראות הכנסת הקלט (צורת ההכנסה לשיקולכם).
3. קלט של ברירת מחדל ושניתן לעריכה.
4. התוצרת של השאלה לאחר הפעלת המתודה ע"י פקד.

ישנם הרבה חבילות של GUI ב Python, הספרייה שמותקנת עם גרסה 3.6 היא Tkinter היא נוחה ,אולם אין הכרח להשתמש דווקא איתה.