תרגיל בית 5

קלט-פלט עם קבצים, טיפול בשגיאות, מילונים ופונקציות

הנחיות כלליות:

- קראו בעיון את השאלות והקפידו שהתכניות שלכם פועלות בהתאם לנדרש.
- את התרגיל יש לפתור לבד!
- הקפידו על כללי ההגשה המפורסמים באתר. בפרט, יש להגיש את כל הפתרונות לשאלות יחד בקובץ ex5_012345678.py המצורף לתרגיל, לאחר החלפת הספרות 012345678 במספר ת.ז. שלכם, כל 9 הספרות כולל ספרת ביקורת.
- מועד אחרון להגשה: כמפורסם באתר.
- בדיקה עצמית: כדי לוודא את נכונותן ואת עמידותן של התוכניות לקלטים שגויים, בכל שאלה הריצו את תוכניתכם עם מגוון קלטים שונים, אלה שהופיעו כדוגמאות בתרגיל וקלטים נוספים עליהם חשבתם (וודאו כי הפלט נכון).
- אין לשנות את שמות הפונקציות והמשתנים שכבר מופיעים בקובץ השלד של התרגיל.
- היות ובדיקת התרגילים עשויה להיות אוטומטית, יש להקפיד על פלטים מדויקים על פי הדוגמאות (עד לרמת הרווח).
- אופן ביצוע התרגיל: שימו לב, בתרגיל זה עליכם להשלים את הקוד בקובץ המצורף.
- **הרצת טסטר:** יחד עם התרגיל קיבלתם קובץ טסטר בשם CodeTests.py. לאחר שפתרתם את כל השאלות ,אתם יכולים להריץ מאותה הספרייה בה סקריפט הפתרון שלכם נמצא. אל תשכחו לשנות את הגדרת הimport ex5_0123456789 בהתאם לשם קובץ הפתרון שלכם. הקובץ כולל מספר בדיקות בסיסיות המוודאות את תקינות הקוד שלכם. אם הבדיקות עברו בהצלחה ,יודפס הטקסט:

```
Non-numeric field encountered in line 1
Inconsistent number of fields detected in line 2
Missing decrypting for code 99
Congrats!!!
All preliminary tests passed!
```

שימו לב ,הטסטר אינו בודק נכונות מלאה של התשובות אלא רק מבצע מספר מצומצם של בדיקות.

- **את התרגיל יש להגיש ללא הטסטר**

שאלה 1

ממשו את הפונקציה **sum_file_nums(infile)** המקבלת בארגומנט **infile** נתיב של קובץ טקסט (**נתיב** הינו משתנה מסוג מחרוזת, המציין את המיקום הייחודי של הקובץ בדיסק, לדוגמא c:/class5/hw5.txt). על הפונקציה לקרוא קובץ טקסט זה המכיל מספר בכל שורה, ולהחזיר את סכום המספרים המופיעים בקובץ (מדפיסה float).
במידה והקובץ ריק יוחזר אפס.
בשאלה זו ניתן להניח שנתיב הקובץ תקין ומציין קובץ קיים, ושכל שורה בקובץ מכילה מספר יחיד שניתן להמרה ל-float. אין צורך לטפל בחריגות.
לדוגמא, עבור הקובץ:

q1_in.txt

המכיל:

```
1.2
2.8
5.0
-1.0
```

יוחזר:

```
8.0
```

במידה והקובץ לא מכיל אף שורה יוחזר:

```
0.0
```

שאלה 2

ממשו את הפונקציה **filter_file_nums(infile, outfile)** המקבלת נתיב לקובץ קלט **infile** המכיל בכל שורה מספר שלם וכותבת לקובץ שנתיבו ניתן על ידי הארגומנט **outfile** את כל המספרים המתחלקים ב-3 לפי סדר הופעתם בקובץ המקורי. בשאלה זו זה ניתן להניח ששמות הקבצים תקינים, שקובץ הקלט קיים ואינו ריק ושכל שורה בו מכילה מספר שלם יחיד. ניתן להניח כי קובץ הפלט לא קיים לפני הריצה של הפונקציה (כלומר, יש ליצור את קובץ הפלט). אין צורך לטפל בחריגות.
לדוגמא, לאחר קריאה לפונקציה עם הפרמטרים הבאים:

filter_file_nums('c:/ q2_in.txt',' c:/ q2_out.txt')

כאשר בתיקייה הנוכחית קיים הקובץ q2_in.txt המכיל:

```
3
5
8
12
14
15
9
```

יופיע בתיקייה הנוכחית קובץ חדש בשם 'q2_out.txt, המכיל את המספרים הבאים:

```
3
12
15
9
```

הערה חשובה: הקפידו על הסימון של שורה חדשה (\n) בסוף על שורה שתכתבו לקובץ הoutfile. אם אין אף שורה לכתיבה, הקובץ outfile יהיה ריק (ולא יכיל את הסימון).

שאלה 3

ממשו את הפונקציה **get_x_freqs(infile, outfile, x)** המקבלת נתיב של קובץ קלט (המחרוזת **infile**), נתיב של קובץ פלט (המחרוזת **outfile**), ומספר שלם וחיובי **x**.
קובץ הקלט מכיל שורות של מילים באנגלית. הפונקציה תכתוב לקובץ הפלט בשורות נפרדות וללא חזרות את רשימת המילים המופיעות לפחות x פעמים בקובץ הקלט בסדר כלשהו. יש להדפיס את המילים באותיות קטנות. במידה ולא קיימת אף מילה בעלת שכיחות העולה או שווה ל-x, יודפס לקובץ "no_words!".

הערות:

- ניתן להניח כי קובץ הקלט infile הוא קובץ טקסט המכיל מילה אחת או יותר בכל שורה כאשר המילים מופרדות על ידי רווחים. כל המילים מורכבות מאותיות באנגלית בלבד.
- אין חשיבות לגודל האותיות (כלומר המילים: dog וDog תיחשבנה לאותה המילה) – יש להדפיס את המילים באותיות קטנות.
- גם תו יחיד יחשב מילה (למשל בצירוף 'a dog', 'a' יחשב כמילה)
- בשאלה זו אין צורך לטפל בשגיאות זמן ריצה.

לדוגמא, אם נפעיל את הפונקציה על קובץ הקלט q3_in.txt (שנמצא בין קבצי התרגיל):

```
The best song in the world
marie had a little lamb but she
stopped being vegetarian on a whim
```

עם x=2, אז היות והמילים 'the' ו-'a' מופיעות פעמיים בקובץ ואילו שאר המילים מופיעות רק פעם אחת, אז בקובץ הפלט יופיע הטקסט הבא:

```
a
the
```

ועבור קובץ הקטסט q3_in_2.txt המכיל את הטקסט:

```
marie had a little lamb but she
```

בקובץ הפלט יופיע הטקסט הבא:

```
no_words!
```

הערה חשובה: שימו לב שהסימון של שורה חדשה (\n) מופיע בסוף כל שורה אותה אתם\ן כותבים לקובץ.

שאלה 4

קובץ CSV (Comma Separated Values file) הוא קובץ טקסט המכיל נתונים במבנה של טבלה (בכל שורה מספר ערכים זהה), כאשר ערכי הטבלה מופרדים על ידי פסיק (ראו שקף 12 במצגת תרגול 5). ניתן לייצג מטריצה באמצעות קובץ CSV כך ששורות המטריצה הן השורות בקובץ, ומספר הערכים בכל שורה בקובץ הוא מספר העמודות במטריצה.
ממשו את הפונקציה **get_csv_matrix(infile)** המקבלת נתיב ל-**infile** וקוראת ממנו מטריצה של מספרים. הפונקציה מחזירה רשימה של רשימות (מטריצה) המכילה את נתוני הקובץ.

על הפונקציה לבדוק שקובץ הקלט הוא אכן קובץ CSV המכיל רק ערכים מסוג int או float. שימו לב כי סוגי ערכים כאלו ניתן להמיר ל-float מבלי שתקפוץ ValueError.

במידה וקובץ הקלט זוהה כקובץ שאינו CSV תקין – כלומר, במידה ומספר הערכים בכל שורה אינו זהה, או שקיימים בקובץ נתונים שאינם ניתנים להמרה ל- float, הפונקציה תחזיר את הערך None.

לדוגמה, קובץ ה-CSV המצורף 'q4_ good.csv ' מכיל את הטקסט הבא:

```
1,3,5,7
2,4,6,8
```

היות וזהו קובץ CSV תקין (מכיל רק מספרים וכל שורה מכילה אותו מספר שדות), ערך ההחזרה צריך להיות הרשימה הבאה:

```
[[1.0, 3.0, 5.0, 7.0], [2.0, 4.0, 6.0, 8.0]]
```

במידה והקובץ איננו תקין, לפני שהפונקציה מחזירה None עליה גם להדפיס הודעת שגיאה המתארת את ההפרה הראשונה **בלבד** בקובץ (מלמעלה-למטה) למסך:

- במידה והתגלתה שורה ובה מס' הערכים שונה ממספר הערכים שנראה עד כה (בשורות הקודמות) תודפס למסך ההודעה:

```
Inconsistent number of fields detected in line #
```

(# ייצג את מספר השורה בה מספר הערכים שונה מהשורה שלפניה, כאשר השורה הראשונה בקובץ תיחשב כשורה מספר 1)

- במידה ואחד האיברים איננו ניתן להמרה לfloat, יש להדפיס למסך:

```
Non-numeric field encountered in line #
```

לדוגמא, עבור הקובץ q4_ bad.csv, ערך ההחזרה צריך להיות None היות והשורה השנייה מכילה מספר שונה ממספר הערכים בשורה הראשונה.

```
1,3,5,7
2,4
```

כמו כן, תודפס למסך ההודעה:

```
Inconsistent number of fields detected in line 2
```

ועבור הקובץ q4_bad2.csv , ערך ההחזרה צריך להיות None היות ואחד הערכים הוא אות שניסיון להמירה ל-float יקפיץ חריגה.

```
1,3,c,7
2,4
```

כמו כן, תודפס למסך ההודעה:

```
Non-numeric field encountered in line 1
```

<u>הערות</u>:

- בבדיקת תקינות קובץ הCSV יש להתעלם מרווחים
- אין חשיבות אם הקובץ מכיל סיומת csv או לא
- ניתן להניח כי הקובץ קיים בדיסק, ואיננו ריק. אין צורך להתמודד עם שגיאות IOError
- במידה ובשורה מסוימת קיימת הפרה של גודל שורה אין לעבד את הספרות המופיעות בה. הדפיסו הודעת שגיאה עבור ההפרה על גדול השורה (ולא על התו הלא חוקי)

הדרכה: היעזרו ב-except כדי לתפוס חריגת ValueError שעשויה להתרחש אם בקובץ מופיע נתון שאינו ניתן להמרה ל-float.

שאלה 5

בשאלה זו נכתוב פונקציה המפענחת הודעה מוצפנת תוך שימוש בקובץ מיפוי.
ממשו את הפונקציה:

decode(input_text_file, code_mapping_file , output_text_file)

המקבלת שלושה נתיבי קבצים:

1. הקובץ input_text_file הינו קובץ הקלט המכיל שורה יחידה הכוללת רצף של מספרים המופרדים ברווחים בלבד. רצף מספרים זה מקודד הודעה סודית אותה נרצה לפענח.
2. הקובץ code_mapping_file הינו קובץ מיפוי המכיל מיפוי של מספרים לתווים (בכל שורה יופיע מספר, פסיק, ותו אליו מקודד המספר). התווים אותם ניתן לקודד כוללים אותיות קטנות באנגלית או קו תחתי בלבד. מספרי הקידוד יכולים להיות כל מספר חיובי ושלם.
3. הקובץ output_text_file הינו קובץ הפלט של הפונקציה, ואליו תיכתב ההודעה המפוענחת.

על הפונקציה לפענח את ההודעה המקודדת על ידי החלפת כל מספר בקובץ הקלט בתו המתאים לו על פי קובץ המיפוי, ולכתוב את הטקסט המפוענח לקובץ הפלט.

לדוגמא:
אם בקובץ הקלט ששמו ניתן ע"י input_text_file מופיע התוכן הבא:

```
1 2 0 10 2 19 22 49 1 41
```

ובקובץ המיפוי code_mapping_file מופיע התוכן הבא:

```
1,s
2,e
0,c
10,r
22,_
41,g
49,m
19,t
```

אז התוכנית תחליף כל מספר בקובץ הקלט, בתו אליו המספר ממופה לפי קובץ המיפוי, ותכתוב את התוכן הבא אל קובץ הפלט output_text_file:

```
secret_msg
```

שימו לב:

- קובץ המיפוי code_mapping_file יכול לכלול מיפויים עבור אותיות קטנות באנגלית ועבור קו תחתי בלבד. ייתכן וקובץ המיפוי יכיל מיפויים רק עבור חלק מהתווים הללו. ניתן להניח כי קובץ המיפוי תקין, וכי לא קיים מספר שממופה לשני תווים שונים.

- עליכם לטעון את קובץ המיפוי למילון, ולאחר מכן להשתמש במילון לטובת פענוח קובץ הקלט. ניתן לכתוב פונקציית עזר שמקבלת את שם קובץ המיפוי, ומחזירה מילון המכיל את המיפויים.

<u>טיפול בשגיאות:</u>

- במידה ואחת מפקודות ה-IO בהן הפונקציה משתמשת זורקת חריגה (exception) מסוג IOError, יש לתפוס את השגיאה, לסגור את כל משאבי הקבצים, להדפיס הטקסט הבא למסך ולצאת מהפונקציה:

```
IO error encountered, cannot decode, exiting!
```

- במידה ובמהלך הריצה על קובץ הקלט, הפונקציה נתקלת במספר שאינו קיים בקובץ המיפוי על הפונקציה לסגור את כל משאבי הקבצים ולזרוק שגיאה מסוג ValueError, הכוללת את הודעת השגיאה הבאה:

```
Missing decrypting for code #
```

שימו לב, הטקסט שתורגם עד למספר לו אין מיפוי צריך להופיע בקובץ הפלט!

כאשר # יוחלף על ידי המספר שמופיע בקובץ הקלט אך אינו מופיע בקובץ המיפוי.

לדוגמא:

אם בקובץ הקלט ששמו ניתן ע"י input_text_file מופיע התוכן הבא:

```
1 22 2 22 1 22 55  49
```

ובקובץ המיפוי code_mapping_file מופיע התוכן הבא:

```
1,a
2,b
22,_
41,o
49,k
```

אז התוכנית תחליף כל מספר בקובץ הקלט על סמך המיפוי, עד שתגיע למספר 55 אשר אין לו מיפוי, ותכתוב את התוכן הבא אל קובץ הפלט output_text_file:

```
a_b_a_
```

והשגיאה תכיל את הודעת הבא:

```
Missing decrypting for code 55
```

<u>הערות חשובה:</u>

1. יש להוסיף את הסימון של שורה חדשה (\n) בסוף ההודעה המתורגמת. אולם, אם התרגום הופסק כתוצאה משגיאה אין צורך להוסיף את הסימון \n.
2. ניתן להניח כי הקובץ input_text_file מכיל שורה אחת המורכבת ממספרים בלבד, ושאם הוא קיים הוא איננו ריק