תרגיל בית 5

הנחיות כלליות:

- קראו בעיון את השאלות והקפידו שהתוכניות שלכם פועלות בהתאם לנדרש.
- את התרגיל יש לפתור לבד!
- הקפידו על כללי ההגשה המפורסמים באתר. בפרט, יש להגיש את כל השאלות יחד בקובץ ex5_012345678.py המצורף לתרגיל, לאחר החלפת הספרות 012345678 במספר ת.ז. שלכם, כל 9 הספרות כולל, ספרת ביקורת.
- אופן ביצוע התרגיל: בתרגיל זה עליכם להשלים את הקוד בקובץ המצורף.
- 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).
- חלק מהשאלות נבדקות באופן אוטומטי. לכן, עליכם לרשום את הקוד שלכם אך ורק במקומות המתאימים לכך בקובץ השלד.
- ניתן להניח כי הקלט שמקבלות הפונקציות תקין (אלא אם נכתב אחרת).
- אין לשנות שמות פונקציות או משתנים שקיימים בקובץ השלד של התרגיל.
- אין למחוק את ההערות שמופיעות בשלד.
- אין להשתמש בקריאה לספריות חיצוניות (אסור לעשות **import**).
- מועד אחרון להגשה: כמפורסם באתר.

שאלה 1-סכימת מספרים המופיעים בקובץ

ממשו את הפונקציה sum_nums(file) המקבלת מחרוזת המציינת שם של קובץ קלט. (file)

הניחו שבתוך הקובץ מופיעה שורה בודדת הכוללת סדרת מספרים שלמים המופרדים על ידי רווח בודד.

על הפונקציה לקרוא את הקובץ ולהחזיר את סכום המספרים המופיעים בו.

עבור קובץ קלט המכיל את השורה הבאה:

4 55 3 67 10

הפונקציה תחזיר את הערך 139.

הערה : בשאלה זו ניתן להניח שהקלט תקין ואין צורך לטפל בשגיאות.

שאלה 2- ספירת שורות

בראיון עבודה לחברה המתמחה בניתוח טקסטים אוטומטי, אתם מתבקשים לממש את הפונקציה count_lines(in_file, out_file) המקבלת שתי מחרוזות המייצגות נתיבים של קבצים. הפונקציה תכתוב לקובץ הפלט out_file את מספר השורות בקובץ in_file.

בשאלה זו ניתן להניח שקובץ הקלט הוא קובץ טקסט תקין, ואין צורך לטפל בחריגות.

- מצורף לתרגיל q4_input_example_1.txt כבדיקה לפונקציה. מומלץ ליצור קבצי בדיקה נוספים למקרי קצה בעצמכם.

דוגמא:

בקובץ q2_input_example_1.txt מופיע הטקסט הבא -

line 1

line 2

line 3 -> thus, your code should write to the output file 3

הפעלת הפונקציה עם קלט in_file שמכיל נתיב לקובץ זה, ונתיב נוסף out_file, תכתוב קובץ חדש, בנתיב שהוזן out_file, שיכיל את הטקסט הבא -

3

שאלה 3- ספירת מופעים

התקבלתם לעבודה, ובתור משימה ראשונה אתם מתבקשים לסווג את אופי תוכן המסמכים בתור שמחים או עצובים באופן אוטומטי. אתם כמובן נלהבים להשתמש בטכניקות המתקדמות ביותר של למידת מכונה, אך ראש הצוות מזכירה לכם שפרקטיקה הנדסית חשובה היא להתחיל עם פתרון פשוט, ולהשתמש בו כדי לבדוק אם ועד כמה פתרון מתוחכם נדרש ועדיף.

אם כן, ראש הצוות מבקשת מכם לממש את הפונקציה simple_sent_analysis(in_file).

פונקציה זו מקבלת הנתיב של קובץ הטקסט שצריך לסווג (משתנה בשם in_file) ומחזירה מילון ובו מספר הפעמים שהופיעה המילה happy ומספר הפעמים שהופיעה המילה sad, שיראה בצורה הבאה: .{'happy':num_happy,'sad':num_sad }

ניתן להניח שהטקסט בקובץ הקלט מכיל רק אותיות באנגלית, מספרים, רווח, את התווים הבאים: !?, -;$%. וכן הוא יכול להכיל מספר שורות.

- הספירה צריכה להיות case insensitive, כלומר happy, Happy, hapPY, HAPPY, כולם נספרים כ- happy
- יש להקפיד לא לספור מילים שמכילות happy או sad. למשל המילה saddle לא תספר כ-.sad

- יש להתעלם מהתווים בתוך הסוגריים (!?, -;$%.). כלומר, הקפידו לספור את happy או sad גם אם מופיע אחד התווים לפניהם או אחריהם. למשל happy? או happy% נספרים כ-happy, אבל hap?py לא נספר כ-.happy

- אם אירעה שגיאת IO יש "לתפוס" אותה, ולסיים את הריצה בצורה מסודרת (ללא קריסה). יש לוודא סגירה של כל הקבצים שנפתחו. יש להחזיר מילון ריק, ואז יש לזרוק את השגיאה הבא מסוג IOError, עם ההודעה: "Cannot encode $in_file due to IO error".

```
>>> simple_sent_analysis("fake.file")
Error: can't find file or read data.
{}
```

כאשר $in_file מציין את שם קובץ הקלט (כמו שקיבלתם בקלט).

- מצורף לתרגיל q3_input_example_1.txt כבדיקה לפונקציה. מומלץ ליצור קבצי בדיקה למקרים אחרים, לרבות מקרי קצה
- הצעה, שימוש במתודות של מחרוזות יכול להקל מאוד על הפתרון. למשל המתודה str.replace(old,new) מאפשרת להחליף כל מחרוזת old המופיעה בstr, במחרוזת new
- רמז: אם תתייחסו לתווים (!?, -;$%.) כאל תווי רווח זה יכול לפשט את הבעיה
- רשות, להעשרה, למי שמתעניין בתחום עיבוד שפה (מחוץ לגבולות הקורס), ניתן אפשר לקרוא עוד על בעיות מסוג ניתוח סנטימנט

דוגמא:

בקובץ q3_input_example_1.txt מופיע הטקסט הבא (הצבעים לצורך הסבר ולא מופיעות בקובץ עצמו) –

,happy hap saddle

sad bla sad s!ad

shimi happy!

עבור הפעלת הפונקציה עם נתיב לקובץ הנ"ל, יתקבל כפלט המילון הבא -

{'happy': 2, 'sad': 2}

הערה: שימו לב כי הפרדה באמצעות התווים (!?, -;$%.) שקולה לרווח ולכן "sad!sad" ייספר כמו פעמיים sad לעומת זאת "sadandsad" למשל לא ייספר בכלל

שאלה 4 -ניתוח נתונים מקובץ

סיווג המסמכים שמימשתם עובד מצוין, ובזכותו החברה קיבלה עבודה מחברת הפקות של סדרות טלוויזיה. חברת ההפקות מעוניינת לדעת האם רווחי יותר להפיק סדרות שמחות, עצובות או ניטרליות. החברה סיפקה תסריטים לכל הסדרות שלהם, וכן את הרווחים מהם. צוות אפליקציה כבר הריץ את האלגוריתם שלכם על כל התסריטים, וסיפק לכם קובץ csv שמכיל את שם הסדרה, כמה הרוויחה, סיווגה (happy, sad, או neutral).

ממשו את הפונקציה calc_profit_per_group(in_file), המקבלת כקלט את הקובץ csv האמור, ומחזירה מילון בו מופיע הרווח הממוצע עבור כל קטגוריה.

במידה ולא מופיעה בכלל אחת מהקבוצות, יופיע במקום הרווח הממוצע 'NA'. זה נחשב מצב תקין ואין להקפיץ שגיאה. ראו דוגמא שניה.

- אם אירעה שגיאת IO יש "לתפוס" אותה, ולסיים את הריצה בצורה מסודרת (ללא קריסה), לא להחזיר דבר, ולזרוק הודעת שגיאה מסוג IOERROR עם ההודעה:
  "Can't load $in_file due to IO error."
  כאשר $in_file יוחלף בשם קובץ הקלט (כמו שקיבלתם בקלט).

```
>>> calc_profit_per_group("fake.file")
Can't load fake.file due to IO Error.
>>>
```

- במידה וסדרה מופיעה יותר מפעם אחת תתקבל שגיאת: ValueError
  'The series $series_name appears more than once.'

  כאשר במקום $series_name יודפס שם הסדרה, ויוחזר מילון ריק. במידה ויש יותר מסדרה אחת שחוזרת על עצמה, מספיק להחזיר את שם הסדרה הראשונה שנתקלים בה. ראו דוגמא שלישית.

- יש לבצע את בדיקות הקלט הבאות:
  - **לוודא** שישנן 3 עמודות
  - עמודה שניה מכילה מספרים בלבד
  - עמודה שלישית מכילה **רק** את הערכים sad/happy/neutral בlowercase

אם אחד מאלה (או יותר) לא מתקיים יש לזרוק (באמצעות raise) שגיאה מסוג ValueError עם הודעת השגיאה:
'Invalid input.'

ראו דוגמא רביעית.

מצורף לתרגיל q4_input_example_1.txt כבדיקה לפונקציה. מומלץ ליצור קבצי בדיקה למקרים אחרים, לרבות מקרי קצה בעצמכם.

דוגמא ראשונה לטקסט הנמצא בקובץ בנתיב in_file:

Descendant Without A Conscience,505.4,happy

Wolf Of The Solstice,30000,sad

Women Of Hope,-4000,neutral

Pirates Of Perfection,65467,neutral

Warriors And Soldiers,-5435,sad

Butchers And Soldiers,76542,sad

World Of The Mountain,6536543,sad

Ruination Of Dusk,-2000,happy

Destroying The Stars,5435,happy

Blinded In My Enemies,765745.5,happy

פלט עבור הקלט הנ"ל:

{'happy': 192421.475, 'sad': 1659412.5, 'neutral': 30733.5'}

דוגמא שניה לטקסט הנמצא בקובץ שבנתיב in_file (כמו הדוגמא הקודמת, רק במחיקת השורות עם neutral) :

Descendant Without A Conscience,505.4,happy

Wolf Of The Solstice,30000,sad

Warriors And Soldiers,-5435,sad

Butchers And Soldiers,76542,sad

World Of The Mountain,6536543,sad

Ruination Of Dusk,-2000,happy

Destroying The Stars,5435,happy

Blinded In My Enemies,765745.5,happy

פלט עבור הקלט הנ"ל:

{'happy': 192421.475, 'sad': 1659412.5, 'neutral': 'NA'}

דוגמא שלישית לטקסט הנמצא בקובץ בנתיב in_file (ההדגשה לצורך הסבר ולא מופיעה כך בטקסט) :

Descendant Without A Conscience,505.4,happy

Wolf Of The Solstice,30000,sad

Women Of Hope,-4000,neutral

Pirates Of Perfection,65467,neutral

Warriors And Soldiers,-5435,sad

Butchers And Soldiers,76542,sad

World Of The Mountain,6536543,sad

Ruination Of Dusk,-2000,happy

Destroying The Stars,5435,happy

Blinded In My Enemies,765745.5,happy

Women Of Hope,-3000,neutral

במקרה זה תתקבל הודעת שגיאה:

ValueError: The series Women Of Hope appears more than once.

ויוחזר מילון ריק {}.

```
>>> calc_profit_per_group("C:\\Users\\LENOVO\\Desktop\\python2020\\hw5\\q4input3.txt")
ValueError: The series '+tkns[0]+' appears more than once.
{}
>>>
```

דוגמא רביעית לטקסט הנמצא בקובץ בנתיב in_file (הצבעים לצורך הסבר ולא מופיעים כך בטקסט) :

Descendant Without A Conscience,505.4,Happy

Wolf Of The Solstice,30000,glad

Women Of Hope,-4000,

Pirates Of Perfection,a lot money,neutral

Warriors And Soldiers,-5435,sad

Butchers And Soldiers,76542,sad

World Of The Mountain,6536543,sad

Ruination Of Dusk,-2000,happy

Destroying The Stars,5435,happy

Blinded In My Enemies,765745.5,happy

בדוגמא זו הקלט לא תקין, ולכן תתקבל הודעת שגיאה:

"ValueError: Invalid input." ולא יוחזר דבר

```
>>> calc_profit_per_group("C:\\Users\\LENOVO\\Desktop\\python2020\\hw5\\q4input4.txt")
Traceback (most recent call last):
  File "<pyshell#55>", line 1, in <module>
    calc_profit_per_group("C:\\Users\\LENOVO\\Desktop\\python2020\\hw5\\q4input4.txt")
  File "C:\Users\LENOVO\Desktop\python2020\hw5\ex5_sol\ex5_sol.py", line 107, in calc_profi
t_per_group
    raise ValueError('Invalid input.')from None
ValueError: Invalid input.
>>>
```

טיפ: על מנת להחזיר את השגיאה בלי שגיאות נוספות שמיוצרות על ידי הפייתון ניתן להשתמש ב:
raise ValueError('Invalid input.') **from None**

שאלה 5 - פיענוח קובץ טקסט מוצפן

ממשו את הפונקציה decode(in_file, out_file) הקוראת טקסט מוצפן מהקובץ in_file , מפענחת אותו על פי החוקיות שתוגדר בהמשך, וכותבת את הטקסט המפוענח לקובץ out_file.
עליכם לפענח את הטקסט שבקובץ הקלט על ידי החלפת כל אות אנגלית באות הקודמת לה (אות גדולה בגדולה וקטנה בקטנה). כל תו שאינו אות באנגלית (רווחים וירידות שורה) יש להשאיר בדיוק כפי שהוא בקובץ הקלט.

לדוגמא, האות B בקובץ הקלט תוחלף באות A שתיכתב במקומה לקובץ הפלט, האות h תוחלף באות g שתיכתב במקומה לקובץ הפלט.

שימו לב האות a תוחלף ב- z והאות A תוחלף באות, Z

לדוגמא, הטקסט המוצפן "Qzuipo Qsphsbnnjoh gps Fohjoffst" יפוענח ל-

"Python Programming for Engineers" .

אם אירעה שגיאת IO במהלך הקריאה או הכתיבה לקבצים יש "לתפוס" אותה ולהדפיס למסך את ההודעה: 'Can't decipher {in_file} due to an IO Error.'

```
>>> decode("fake.file","fake.file")
Cannot decipher fake.file due to an IO Error.
>>>
```

כאשר עליכם להחליף את {in_file} בערך של שם הקובץ כאשר מחזירים את ההודעה.
במקרה זה יש לצאת מהתוכנית בצורה מסודרת ולנסות לסגור את הקבצים בטרם היציאה.

הקובץ q5.txt מכיל טקסט מוצפן. אם תבצעו את הפענוח נכון, התוצאה תהיה זהה לתוכן הקובץ q5_deciphered.txt

שימו לב:

- ניתן להניח שקובץ הקלט כולל אותיות גדולות או קטנות באנגלית, רווחים, וסימן ירידת שורה בלבד.סימן ירידת שורה בחלונות ובלינוקס שונה (חלונות \n -לינוקס - \r\n)
- יש לוודא שחרור משאבים על ידי סגירה מסודרת של הקבצים גם אם היתה שגיאה. רמז: השתמשו ב- finally .
- ניתן להשתמש בפוקנציית ord בשאלה על מנת להשוות בין אותיות ובפונקציה chr כדי להמיר מספרים לאותיות
- עבור קובץ שלא קיים – לדוגמה 'not_exist.txt' יודפס למסך– 'Cannot decipher not_exist.txt due to an IO Error.'