תכנות בפייתון למדעים דיגיטליים בהייטק, סמסטר א' תשפ"א 2020-2021

תרגיל בית 5

קלט-פלט (I/O) וחריגות

הנחיות כלליות:

- קראו **היטב** את השאלות והקפידו שהתכניות 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 הביקורת.
- 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 וכי התוכנית אינה קורסת).
- היות ובדיקת התרגילים עשויה להיות אוטומטית, **יש להקפיד על פלטים מדויקים על פי הדוגמאות (עד לרמת הרווח).**
- אופן ביצוע התרגיל: בתרגיל זה עליכם להשלים את הקוד בקובץ המצורף.
- **אין לשנות את שמות המשתנים שכבר מופיעים בקובץ השלד של התרגיל.**
  יש לעבוד עם המשתנים שמופיעים בשלד התרגיל. על הקוד של כל שאלה לעבוד ולספק את התוצאה הדרושה עבור קלט שיוזן במשתנים שמופיעים בשלד (המשתנים שלידם סימני שאלה ומחכים לקלט כפי שראינו בדוגמא מהתרגול). יחד עם זאת, אתם רשאים להוסיף משתנים נוספים כראותם עינכם.
- **שימו לב מתי מבקשים מכם להחזיר ערך מפונקציה (return) ומתי מבקשים להדפיס למסך (print).**
- אין למחוק את ההערות שמופיעות בשלד.

שאלה 1 (קריאה מקובץ):

ממשו את הפונקציה ***max_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 המספר הגדול ביותר בסדרת המספרים המופיעים בו.
לדוגמא, כאשר נקרא לפונקציה כך:

```
>>> max_nums('q1.txt')
```

אז עבור קובץ הקלט q1.txt המצורף לתרגיל ומכיל את השורה הבאה:

```
4 55 3 67 10
```

הפונקציה תחזיר את הערך 67.

הערה: בשאלה זו ניתן להניח שהקלט תקין, שהקובץ נמצא בתיקיה הנוכחית ואין צורך לטפל בשגיאות.
מותר להשתמש בפונקציות מובנות של פייתון.

שאלה 2 (קריאה וכתיבה לקבצים):

ממשו את הפונקציה ***copy_capitalized_words (infile, outfile)*** אשר מעתיקה מילים המתחילות באות גדולה בלבד מקובץ הקלט infile אל קובץ פלט בשם outfile.

לדוגמא, כאשר נקרא לפונקציה כך:

```
>>> copy_capitalized_words ('q2.txt', 'q2_out.txt')
```

אז עבור קובץ הקלט q2.txt המצורף לתרגיל ומכיל את השורות הבאות:

```
hello
TAU
hello
Python
```

הפונקציה תיצור קובץ חדש q2_out.txt שיכיל את השורות הבאות:

```
TAU
Python
```

שאלה 3 (חישוב שכיחויות מילים בקובץ):

ממשו את הפונקציה **get_x_freqs (infile, outfile, x)** המקבלת שם של קובץ קלט (המחרוזת **infile**), שם של קובץ פלט (המחרוזת **outfile**) ומספר שלם חיובי **x**.

הפונקציה תכתוב לקובץ הפלט בשורות נפרדות וללא חזרות את x המילים השכיחות ביותר בקובץ הקלט ואת מספר המופעים של כל אחת מהן. על המילים להיכתב כשהן ממוינות בסדר יורד לפי שכיחות ההופעה שלהן בקובץ הקלט. רווח בודד יפריד בכל שורה בין המילה לבין מספר המופעים שלה.

הערות:

- קובץ הקלט infile הוא קובץ טקסט המכיל מילה אחת או יותר בכל שורה כאשר המילים מופרדות על ידי רווחים (רווח בודד בין מילים וירידת שורה בין שורות). כל המילים מורכבות מאותיות קטנות בלבד.
- בשורה הראשונה בקובץ הפלט outfile תופיע המילה השכיחה ביותר בקובץ הקלט infile. במידה ויש כמה מילים המופיעות באותה שכיחות, אין חשיבות לסדר דירוגן הפנימי בקובץ הפלט.
- על התוכנית לשמור לקובץ הפלט בדיוק x מילים. במידה וישנן כמה מילים עם אותו מספר מופעים, יתכן ורק חלק מהמילים תישמרנה לקובץ (כדי לא לחרוג מ-x). במצב כזה אין חשיבות לבחירת המילים מתוך אוסף המילים בעלות אותו מספר מופעים.
- במידה והמחרוזות infile או outfile הן ריקות, יש להעלות שגיאה מסוג ValueError (יש להשתמש ב-raise) הכוללת את הודעת השגיאה: 'Invalid file name' . במידה והמחרוזות אינן ריקות, ניתן להניח שהקלט תקין.
- הדרכה: היעזרו במילון לחישוב שכיחויות המילים כפי שראינו בדוגמאות בכיתה.

לדוגמא, אם נפעיל את הפונקציה על קובץ הקלט ('q3.txt') שנמצא בין קבצי התרגיל ונגדיר ש-x=3, אז היות והמילה "round" מופיעה 8 פעמים בקובץ הקלט, המילה "the" מופיעה 5 פעמים והמילה "and" מופיעה 4 פעמים, אז בקובץ הפלט יופיע הטקסט הבא:

```
round 8
the 5
and 4
```

שימו לב שאין חשיבות לסדר הפנימי של הופעת מילים בקובץ הפלט אם יש להן אותו מספר מופעים בקובץ הקלט. לדוגמא, אם נפעיל את הפונקציה על קובץ הקלט ('q3.txt') שנמצא בין קבצי התרגיל ונגדיר ש-x=4,אז היות ולכל המילים הבאות :

Wheels, go, bus, on

יש מספר מופעים שווה של 2 בקובץ הקלט, אז יהיו לנו מספר אפשרויות לתשובה נכונה בקובץ הפלט :

```
round 8
the 5
and 4
wheels 2
```

Or

```
round 8
the 5
and 4
go 2
```

Or

```
round 8
the 5
and 4
on 2
```

Or

```
round 8
the 5
and 4
bus 2
```

שאלה 4 (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.

את הפענוח יש לבצע ע"פ קידוד ASCII [לחיצה תוביל לטבלת ASCII] המגדיר לכל תו ערך מספרי כלשהו.

עליכם לפענח את הטקסט שבקובץ הקלט על ידי החלפת כל אות אנגלית באות **הקודמת** לה. 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 a תוחלף ב-z, האות A תוחלף באות Z, והאות H תוחלף באות G שתיכתב במקומה לקובץ הפלט.

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 file due to an IO Error. '

במקרה זה יש לצאת מהתוכנית בצורה מסודרת ולנסות לסגור את הקבצים בטרם היציאה.

הקובץ q4.txt המצורף לתרגיל מכיל טקסט מוצפן. אם תבצעו את הפענוח נכון, התוצאה תהיה זהה לתוכן הקובץ q4_deciphered.txt שמצורף אף הוא.

שימו לב:

- ניתן להניח שקובץ הקלט כולל אותיות גדולות או קטנות באנגלית, רווחים, וסימן ירידת שורה בלבד.
- יש לוודא שחרור משאבים על ידי סגירה מסודרת של הקבצים גם אם היתה שגיאה. רמז: השתמשו ב-finally.

שאלה 5 (עיבוד קובץ נתונים במבנה טבלאי):

קובץ CSV הוא קובץ טקסט המכיל נתונים במבנה של טבלה מלבנית כאשר פסיקים משמשים כתו מפריד בין השדות בכל שורה (ראו מצגת תירגול 5).

ממשו את הפונקציה **process_contacts(contacts_file)** המקבלת כקלט שם של קובץ csv המכיל טבלה המתארת את פרטיהם של אוסף אנשי קשר. כל שורה תייצג איש קשר ותכלול 4 שדות המופרדות על ידי פסיק: שם פרטי, שם משפחה, כתובת, ועיר מגורים. הפונקציה תקרא את הקובץ, תעבד את הנתונים המאוחסנים בו ותחזיר מילון שממפה לכל שם עיר את רשימת שמות המשפחה של תושביה (ללא חזרות).

- על הפונקציה להתעלם משורות הערה אשר מתחילות בסולמית ('#').
- שימו לב שהעמודה השנייה מציינת שם משפחה של איש הקשר והעמודה הרביעית מציינת את עיר המגורים שלו.
- על הפונקציה לבדוק שקובץ הקלט תקין, כלומר שבכל שורה יש 4 שדות, ושאף שדה אינו ריק. במידה וקובץ הקלט זוהה כקובץ שאינו תקין יש להעלות שגיאה (באמצעות raise) מסוג ValueError הכוללת את הודעת השגיאה 'Invalid input file'. ניתן להניח שמלבד בעיות אלו, הקובץ קיים ותקין.
- במקרה של שגיאת IOError, יש לתפוס את השגיאה, להדפיס למסך את הודעת השגיאה 'IO Error encountered', לסגור את הקובץ, ולהחזיר מילון ריק.

לדוגמה ,עבור הקובץ המצורף , 'q5_good.csv' שהוא קובץ CSV תקין המכיל נתונים על אנשי קשר לפי הפורמט שקבענו

```
Avi,Levi,Kushnir 7,Jerusalem
Moshe,Yarden,Hamakabim 4,Tel Aviv
Michael,Cohen,Herzel 70,Tel Aviv
#This is a comment
Eli,Cohen,Haroe 6,Jerusalem
Moti,Cohen,shalom 5,Tel Aviv
```

הפונקציה תחזיר את המילון הבא המכיל שני מפתחות (סדר המפתחות או האיברים ברשימה אינו משנה):

{'Jerusalem' : ['Levi', 'Cohen'], 'Tel Aviv' : ['Cohen', 'Yarden']}

עבור הקובץ 'q5_bad.csv' שמכיל את הטקסט הבא:

```
Avi,Levi,Kushnir 7,Jerusalem
Moshe,,Hamakabim 4,Tel Aviv
Michael,Cohen,Herzel 70
Eli,Cohen,Haroe 6,Jerusalem
Moti,Cohen,shalom 5,Tel Aviv
```

הפונקציה תעלה שגיאה כי שם המשפחה בשורה השניה חסר, וגם כי מספר השדות בשורה השלישית שונה מ-4.

שאלה 6 (עיבוד קובץ נתונים גדול):

באתר מאגרי המידע הממשלתיים data.gov.il מפרסם משרד הבריאות נתונים עדכניים לגבי מספר הנדבקים בוירוס הקורונה באיזורים גיאוגרפיים שונים בארץ החל מתחילת המגיפה.

הקובץ `geographic-summary-per-day-2020-11-18.csv` שנמצא בין קבצי התרגיל, הורד מהקישור הבא:

https://data.gov.il/dataset/covid-19/resource/d07c0771-01a8-43b2-96cc-c6154e7fa9bd

הסבר על מבנה הקובץ מובא כאן:

https://data.gov.il/dataset/f54e79b2-3e6b-4b65-a857-f93e47997d9c/resource/22d5dad8-e0ef-425b-86f9-655fbe468823/download/geographical-distribution-readme.pdf

הקובץ מציין בין היתר את מספר הנדבקים המצטבר (accumulated_cases, עמודה #6) עבור תאריך מסוים (date, עמודה #3) ויישוב מסוים (town, עמודה #14). שימו לב שיישובים גדולים מחולקים לאיזורים גיאו סטטיסטיים (עמודה #2), ולכן עבור יישובים אלו תופענה כמה שורות עבור כל תאריך.

פיתחו את הקובץ קודם בתוכנת גיליון אלקטרוני (כגון EXCEL) , ואח"כ בעורך טקסט (כגון notepad++) והתרשמו מהמבנה שלו.

ממשו את הפונקציה **get_covid_cases_by_date(filename, date)** אשר מקבלת מחרוזת filename המציינת את שמו של קובץ הנתונים, ומחרוזת date המציינת תאריך כלשהו. הפונקציה תדפיס למסך את 10 היישובים בעלי מספר המקרים הגבוה ביותר בתאריך המצוין כאשר שמות היישובים ממוינים בסדר יורד לפי מספר המקרים.

שימו לב:

- בפלט יופיע בכל שורה מספר המקרים, טאב בודד ואז שם היישוב.

- במידה ועבור תאריך (עמודה 3#) ושם יישוב מסוים (עמודה 14#) מופיעות כמה שורות שונות (אם היישוב מחולק לכמה איזורים גיאו סטטיסטיים) אז יש לסכום את מספר הנדבקים בשורות השונות כדי לקבל את מספר הנדבקים המצטבר הכולל ביישוב.
- היות וקובץ הנתונים כולל תווים בעברית, הקפידו לפתוח את הקובץ תוך ציון הקידוד UTF-8 (הקידוד מגדיר איך מתורגמים הביטים בקובץ לתווים), למשל כך:

```
f = open(filename, 'r', encoding='UTF-8')
```

- מומלץ להשתמש במילון כדי לשמור את מספר המקרים המצטבר עבור כל יישוב בתאריך המבוקש.
- במידה והקובץ אינו מכיל נתונים עבור התאריך המבוקש, יש להדפיס הודעת שגיאה כמתואר בדוגמת ההרצה השלישית.
- במידה והפונקציה נתקלת בחריגת IOError , יש להדפיס למסך IO Error encountered ולצאת בצורה מסודרת מהתוכנית לאחר סגירת קובץ הקלט.
- שימו לב שהעמודה המתארת את מספר המקרים המצטבר מכילה את המחרוזת '<15' עבור ערכים הקטנים מ-15. במקרים אלו נניח שמספר המקרים הוא 0.

דוגמאות הרצה:

```
>>> get_covid_cases_by_date('geographic-summary-per-day-2020-11-18.csv', '2020/04/01')
890  בני ברק
381  ירושלים
75   אלעד
59   אפרת
55   מגדל העמק
52   אשקלון
49   מודיעין-מכבים-רעות
45   יהוד
43   בית שמש
37   "כפר חב""ד"
```

```
>>> get_covid_cases_by_date('geographic-summary-per-day-2020-
11-18.csv', '2020/05/18')
2970 ירושלים
2895 בני ברק
481  בית שמש
418  מודיעין עילית
371  אלעד
280  ביתר עילית
152  דייר אל-אסד
149  חורה
119  אשקלון
109  אור יהודה

>>> get_covid_cases_by_date('geographic-summary-per-day-2020-
11-18.csv', '2020/11/15')
43564     ירושלים
25624     בני ברק
12493     אשדוד
10979     מודיעין עילית
9420 תל אביב-יפו
7828 בית שמש
7403 פתח תקווה
6830 נתניה
5956 ביתר עילית
5666 באר שבע

>>> get_covid_cases_by_date('geographic-summary-per-day-2020-
11-18.csv', '2025/15/15')
No data is available for date 2025/15/15
```

בהצלחה !