תכנות בפייתון למדעים בהייטק 2020, סמסטר א' תשפ"א

תרגיל בית 6

חזרה על ניתוח קבצי טקסט

רקורסיה

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 ex6_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.

חלק א' – ניתוח קבצי נתונים

שאלה 1:

בשאלה זו ננתח קובץ טקסט בפורמט CSV המכיל נתונים מאתר המלצות הסרטים IMBD.

לקבצי התרגיל מצורף הקובץ IMDB-Movie-Data.csv אשר מכיל מידע לגבי 1000 הסרטים הכי פופולריים באתר בשנים 2006-2016.

הקובץ נלקח מאתר Kaggle (https://www.kaggle.com/PromptCloudHQ/imdb-data)

- השורה הראשונה בקובץ הנתונים היא שורת כותרת המכילה את התיאור של כל עמודה. כל שורה בקובץ החל מהשורה השניה מייצגת מידע על סרט מסוים.
- כל עמודה בקובץ מייצגת מאפיין של הסרט, לפי הפירוט המופיע בטבלה שבהמשך.
- שימו לב שהשדות בכל שורה מופרדים על ידי פסיק (היות ומדובר בקובץ CSV). במידה ובתוך שדה מסוים יש כמה ערכים שונים, הם יהיו מופרדים על ידי נקודה פסיק (;).
- רוב הסרטים משתייכים ליותר מז'אנר אחד.
- מומלץ לפתוח קודם כל את הקובץ בתוכנת Excel כדי להתרשם מהמבנה שלו.

תיאור עמודות הקובץ:

Column	Description
Rank	Movie rank order
Title	The title of the film
Genre	A comma-separated list of genres used to classify the film
Description	Brief one-sentence movie summary
Director	The name of the film's director
Actors	A comma-separated list of the main stars of the film
Year	The year that the film released as an integer.
Runtime (Minutes)	The duration of the film in minutes.
Rating	User rating for the movie 0-10
Votes	Number of votes
Revenue (Millions)	Movie revenue in millions
Metascore	An aggregated average of critic scores. Values are between 0 and 100. Higher scores represent positive reviews.

ממשו את הפונקציה **top5_by_genre(genre_name, file_name)** אשר מקבלת מחרוזת המציינת שם ז'אנר (Genre), ומחרוזת המכילה את שם קובץ נתוני הסרטים. הפונקציה תכתוב לקובץ בשם top5.csv את נתוניהם של 5 הסרטים מהז'אנר הנתון שהינם בעלי ההכנסה הגבוהה ביותר (Revenue). בנוסף לכתיבת קובץ הפלט, הפונקציה גם תחזיר ערך המציין את סטטוס הריצה שלה: הערך 1 יציין שהפונקציה רצה בהצלחה, והערך 0 ייצג שהפונקציה נתקלה בבעיה ולא סיימה לרוץ בהצלחה.

הבהרות:

- קובץ הפלט יכיל את שורת הכותרות בדומה לקובץ הקלט. ולאחר שורת הכותרת תופענה עד 5 שורות המכילות את כל פרטי הסרטים הרלבנטיים כולל כל הפרטים שמופיעים עליהם בקובץ הקלט.
- במידה והפונקציה נתקלת בשגיאת IO כלשהי (למשל – קובץ הקלט אינו קיים או אם לא ניתן לכתוב לקובץ הפלט) אז הפונקציה תדפיס למסך את ההודעה "IO Error encountered" ותחזיר 0.
- שימו לב שגודל האותיות אינו משנה כאשר משווים את שם הז'אנר הנתון לשמות הז'אנרים המופיעים בקובץ (Case Insensitive).

דוגמת הרצה:

```
>>> top5_by_genre("Comedy", "file_that_does_not_exist.csv")
IO Error encountered
0
>>> top5_by_genre("action", "IMDB-Movie-Data.csv")
1
```

קובץ הפלט עבור דוגמא זו מצורף, שמו top5.csv.

חלק ב' – רקורסיות קלות

שאלה 2 :

ממשו את הפונקציה הרקורסיבית **mult(x, y)** המקבלת שני מספרים שלמים וחיוביים x ו-y ומחזירה את מכפלתם זה בזה מבלי להשתמש בפעולת הכפל (האופרטור *) או לולאות.

דוגמאות הרצה:

```
>>> mult(4, 5)
20
>>> mult(2, 3)
6
```

ניתן להניח ש-x ו-y הם מספרים שלמים, חיוביים ושונים מאפס.

שאלה 3:

ממשו את הפונקציה הרקורסיבית **count_val(lst, val)** המקבלת רשימה lst וערך val ומחזירה את מספר המופעים של val ברשימה lst.

אין להשתמש בלולאות או בפונקציות מובנות של פייתון לספירת איבר ברשימה.

דוגמאות הרצה:

```
>>> count_val([6,5,5,2,5,3],5)
```

```
3
>>> count_val([1,2,3],5)
0
>>> count_val([],5)
0
>>> count_val(list('Enterprise'),'r')
2
```

חלק ג' – רקורסיות קצת פחות קלות

שאלה 4:

אתם מטפסים במעלה גרם מדרגות בן n (מספר שלם וחיובי) מדרגות. בכל צעד של טיפוס אתם יכולים לבחור לעלות מדרגה אחת בלבד או שתי מדרגות בבת אחת. בכמה דרכים שונות ניתן לטפס לקצה גרם המדרגות על ידי צירופים שונים של מדרגה אחת או שתיים ?

דוגמאות :

- יש אפשרות אחת לטפס גרם של מדרגה אחת (צעד בודד של מדרגה אחת)
- יש אפשרות אחת לטפס גרם של 2 מדרגות (צעד בודד של 2 מדרגות)
- יש 3 אפשרויות לטפס גרם של 3 מדרגות (3 צעדים של 1, צעד של 1 ואז צעד של 2 או צעד של 2 ואז צעד של 1. שלוש האפשרויות ניתנות לייצוג תמציתי כ- 111,12,21)
- יש 8 אפשרויות לטפס גרם של 5 מדרגות (11111,1112,1121,1211,122,2111,212,221)

סעיף א'

ממשו את הפונקציה **הרקורסיבית climb_combinations(n)** אשר מקבלת מספר שלם וחיובי (גדול מאפס) n ומחזירה את מספר האפשרויות השונות להגיע לקצה גרם המדרגות .

הערה: המימוש שתכתבו לפונקציה זו ייבדק עבור ערכי n הקטנים מ-40 (בסעיף הבא תעשו שימוש בממואיזציה כדי להתגבר על מגבלה זו).

דוגמאות הרצה :

```
>>> climb_combinations(5)
8
>>> climb_combinations(10)
89
```

סעיף ב'

ממשו את הפונקציה **הרקורסיבית climb_combinations_memo(n, memo=None)** אשר בדומה לסעיף א' מקבלת את n ומחזירה את מספר האפשרויות לעלות את גרם המדרגות, אך הפעם השתמשו **בממואיזציה** על מנת לשפר את זמן הריצה של הרקורסיה.

שימו לב שהפלטים של גרסה זו של הפונקציה אמורים להיות זהים לפלטים של הפונקציה מסעיף א' עבור אותם ערכי n.

הערה: המימוש שתכתבו לפונקציה זו ייבדק גם עבור ערכי n הגדולים מ-40, וזמן הריצה עבור n<1000 אמור להיות נמוך מ-3 שניות.

דוגמאות הרצה:

```
>>> climb_combinations(10)
89
>>> climb_combinations(100)
573147844013817084101
```

שאלה 5:

אסטרונאוט שחלליתו נפגעה ממטאוריט נדרש להשליך מסיפון החללית פריטים אישיים שסכום משקלם הוא בדיוק W קילוגרם על מנת להיכנס חזרה למסלול בטוח לכדור הארץ.

ממשו פונקציה **רקורסיבית can_return_to_earth(weights, W, K)** שתקבל את רשימת המשקלים של הפריטים האישיים שהאסטרונאוט לקח עמו לחלל ואת (weight) ואת המשקל הכולל שיש להשליך מהחללית (W) , והחזירו True אם קיים אוסף פריטים ברשימה שסכום משקלם הוא בדיוק W ו-False אחרת.

שימו לב: האסטרונאוט יכול להשליך רק פריטים שמשקלם הבודד קטן מ-K קילוגרם היות ומנוף החללית נפגע ואינו מתפקד.

- ניתן להניח ש K וכל האיברים ברשימת המשקלים הם מספרים שלמים חיוביים (גדולים מ-0) וש-W אינו שלילי
- ניתן לבחור כל פריט רק פעם אחת
- אין להשתמש בלולאות או בפונקציות מיון מובנות של פייתון

דוגמא:

```
>>> can_return_to_earth ([5,2,6,4,2],8,5)
True
```

הסבר: הפונקציה החזירה True כי קיים ברשימה אוסף של איברים הקטנים מ-5 שסכומם הוא 8.

```
>>> can_return_to_earth ([5,2,6,4,2],8,3)
 False
```

הסבר : הפונקציה החזירה False כי לא קיים ברשימה אוסף של איברים הקטנים מ-3 שסכומם הוא 8.

בהצלחה !