הנחיות כלליות

יש לשלוח את הקבצים באמצעות מערכת ההגשה לפני חלוף התאריך 28/1/20.

ניתן להגיש את התרגיל באיחור עם קנס אוטומטי על פי הפירוט הבא:

- יום איחור - קנס של 5 נקודות (ציון מקסימלי 95).
- יומיים איחור - קנס של 15 נקודות (ציון מקסימלי 85).
- שלושה ימי איחור - קנס של 30 נקודות (ציון מקסימלי 70).

לאחר מכן לא יהיה ניתן להגיש את התרגיל (ציון 0).

המתרגל האחראי על התרגיל הוא יואב.

שאלות בנוגע לתרגיל יש לפרסם **באופן ציבורי בפורום הקורס** בלבד! רק אם לא התקבלה תשובה לאחר 24 שעות, יש לשלוח מייל לכתובת yoavgilor@gmail.com עם קישור לדיון הרלוונטי.

בקשות להארכה מסיבות מוצדקות (מילואים, לידה וכו') יש לפרסם **באופן פרטי בפורום הקורס** בלבד (יש למען את הפוסט ל-instructors). בכל בקשה יש לציין שם מלא, שם משתמש במערכת ההגשה, תעודת זהות והאם אתם ממדעי המחשב או מתמטיקה.

יש להקפיד מאוד על הוראות עיצוב הקלט והפלט, בדיוק על פי הדוגמאות המצורפות.
שימו לב להנחיות במסמך ה-Coding Style המפורסם באתר הקורס.

עליכם לכתוב קוד על פי ההנחיות ולוודא שקיבלתם 100 בבדיקה האוטומטית הראשונית, וכן שהתרגיל מתקמפל ורץ על שרתי המחלקה (u2) ללא שגיאות או אזהרות.
תרגיל שלא עומד בסטנדרטים הבסיסיים הללו יגרור **ירידה משמעותית בציון התרגיל**, בשל הטרחה שהוא מייצר בתהליך הבדיקה שלו.

להזכירכם העבודה היא אישית. "עבודה משותפת" דינה כהעתקה. התרגיל נבדק על ידי מערכת ההגשה האוטומטית גם מהבחינה הזו, ותרגיל שהועתק יגרור ציון 0 **לכל הגורמים** השותפים בהעתקה. אתם יכולים לדון בגישות לפתרון התרגיל באופן תיאורטי, אך אין לשתף קוד בשום צורה.

בפיתוח הקוד ניתן להשתמש בכל סביבת עבודה, העיקר הוא שתדעו איך לקחת את קבצי הקוד מתוך הסביבה הזו, לבדוק אותם על שרתי האוניברסיטה ולהגיש אותם באמצעות מערכת ההגשה.
דוגמאות לחלק מהסביבות האפשריות:

IDEs (Integrated Development Environment):

- PyCharm
- Anaconda / Spyder
- Visual studio
- Clion
- Eclipse
- Xcode

Text Editors:

- Atom
- Sublime
- Notepad++
- Vim

בהצלחה!

תרגיל 7 – ass7

משקל התרגיל מתוך ציון התרגול: 15%.

הסברים

בתרגיל זה עליכם לבנות זחלן (Crawler). זחלן הוא כלי לסריקת האינטרנט על ידי שימוש בקישורים המופיעים בעמודי האינטרנט השונים. הזחלן "גולש" באינטרנט, "לוחץ" על כל הלינקים שהוא "רואה", ושומר את המידע. דרך זו אוספת מידע על הקישורים שיש באינטרנט ומשמשת חברות כמו Google בהערכת אילו עמודי אינטרנט יותר חשובים מאחרים.

עמודי אינטרנט כתובים בשפת html ובשפות אחרות. קובץ html גם הוא קובץ טקסט פשוט והוא מכיל את המידע לגבי מה שמכיל העמוד.

קישור (או: הפניה, לינק) הוא מקום בעמוד אינטרנט שאפשר ללחוץ עליו וזה יעביר אותנו לעמוד אינטרנט אחר. בשפת html הדרך לתאר קישור היא, לדוגמא כך:

```
<a href="page.html">Go to page!</a>
```

פקודה זו (נקראת hyperlink element) כתובה כטקסט בקובץ ה-html והיא תתבטא בעמוד האינטרנט בתור הכיתוב "Go to page" שאם נלחץ עליו, נעבור לעמוד html אחר בשם page.html.

הכיתוב page.html מסמל קובץ שאליו הקישור מפנה. (כאשר אומרים שעמוד א' מפנה לעמוד ב' הכוונה שיש בקוד של עמוד א' פקודת hyperlink כזו, אשר הhref שלה הוא שמו של עמוד ב').

התרגיל

עליכם לכתוב תכנית שתזחל על קבצי html באופן רקורסיבי ותשמור את המידע על הקישורים שיש בהם. מצורפים 7 קבצי html עם קישורים אחד לשני לצורך ניסיון. על התכנית שלכם לבקש מהמשתמש קובץ קלט והמשתמש יקליד שם של קובץ, לדוגמא : '1.html':

```
enter source file:
1.html
```

קובץ זה הוא הקובץ שממנו מתחילים לזחול.

התכנית תפתח את הקובץ ותשלוף את תוכנו לתוך מחרוזת.

התכנית תמצא את המקומות שבהם כתוב "href=" ותשלוף שמות כל הקבצים ש1.html מפנה אליהם ותשמור אותם ברשימה במילון.

התכנית תבצע את אותה פעולה עבור כל הקבצים שנמצאו אלא אם כן הפעולה כבר בוצעה עבור הקובץ הזה (או שבעמוד אין הפניות).

לצורך כך צרו מילון המשמש ל memoization כאשר המפתחות הן שמות הקבצים והערכים הם רשימות הקבצים שאליהם מפנה המפתח.

המילון (עבור הקבצים המצורפים) אמור להראות בערך כך (למעט סדר האיברים):

```
{'1.html': ['2.html', '3.html'], '2.html': ['3.html', '4.html'], '3.html': ['5.html',
'7.html'], '5.html': [], '7.html': ['2.html'], '4.html': ['6.html'], '6.html': ['2.html']}
```

בסוף הסריקה התכנית תשמור את המידע לקובץ בשם results.csv, לדוגמא, עבור 7 הקבצים המצורפים, אמור להישמר הקובץ הבא (אין חשיבות לסדר האיברים):

```
1.html,2.html,3.html
2.html,3.html,4.html
3.html,5.html,7.html
5.html
7.html,2.html
4.html,6.html
6.html,2.html
```

הבינו איך המילון הזה וקובץ ה-csv מעלה למעשה מתארים שניהם את אותו המידע לגבי הקבצים המצורפים.

בקובץ הCSV, בכל שורה כתוב ראשית שם של קובץ ואחריו (מימינו) את הקבצים שאליהם הוא מכיל קישורים.

לדוגמא:

- השורה הראשונה מתארת את הקובץ 1.html והוא מכיל 2 הפניות 2.html ו- 3.html .
- השורה האחרונה מתארת את הקובץ 6.html ויש בו הפניה אחת לעמוד 2.html

נסו לפתוח את עמודים 1 ו-6 בדפדפן וראו שאכן אלו הקישורים שמופיעים בקובץ.

נסו לפתוח אותם גם בעורך טקסט פשוט כדי לראות את ה- hyperlink element שמתאר את קישורים אלו.

לאחר שמירת הקובץ התכנית תבקש מהמשתמש להקליד שם קובץ נוסף:

```
enter file name:
3.html
```

התכנית תדפיס את הרשימה הממוינת (לפי א'-ב') של כל הקבצים אליהם 3.html מפנה:

```
['5.html', '7.html']
```

הערה מכיוון שיש לכם מילון שבו כל מפתח הוא שם של קובץ והערך הוא אותה רשימה, ניתן לשלוף אותה באמצעות המפתח, ואז ניתן למיין ולהדפיס אותה באמצעות פקודות פשוטות.

מומלץ לכתוב שלוש פונקציות:

1. פונקציית הזחלן המתוארת מעלה.
2. פונקציית parser פונקציה שרצה על מחרוזת ה-html ומוצאת את הטקסט: "href=" ושומרת את מה שיש אחריו, עד למרכאות.

דוגמא:

```
enter source file:
1.html
enter file name:
3.html
['5.html', '7.html']
```

הערות

- ניתן להניח שכל הלינקים מסומנים ב=href ועם מרכאות כפולות (" כן. ' לא).
- הבדיקה תעשה מספר פעמים ועם קבצים שונים מאלו שנתונים כאן.
- ניתן להניח שהקלט בתכנית כולה יהיה תקין.
- על ההדפסות להיות זהות להדפסות שמתוארות מעלה.
- קובץ הcsv צריך להיות ללא רווחים וללא שורות רווח.
- יש להגיש קובץ אחד בלבד בשם ass7.py

המלצות:

- מומלץ מאוד להשתמש בdictionary עבור memoization.
- למיון רשימה בשם list נכתוב את הפקודה list.sort().
- לקלט מהמשתמש: פקודת str=input() תקלוט קלט מהמשתמש לתוך משתנה מסוג מחרוזת בשם str.
- ניתן להדפיס רשימה בשם list עם הפקודה print(list) והיא תודפס כמו שנדרש בתרגיל זה.