בית הספר להנדסה ומדעי המחשב ע"ש רחל וסלים בנין

מבוא למדעי המחשב 67101 – סמסטר א' 2020/2021
תרגיל 5 – רשימות ושימוש בקבצים
להגשה בתאריך **26/11/2020** בשעה 22:00

הקדמה

בתרגיל זה נתרגל שימוש בלולאות, רשימות רב מימדיות, מילונים, פרמטרים בשורת פקודה ועבודה עם קבצים. אנו ממליצים להתחיל לעבוד על התרגיל בשלב מוקדם שכן התרגיל ארוך מקודמיו.

מטרת התרגיל היא כתיבת תכנית אשר מוצאת מילים בתוך מטריצה של אותיות (תפזורת). התכנית תקבל כחלק משורת הפקודה את הפרמטרים הבאים:

1. שם קובץ המילים - המכיל רשימה של מילים תקניות אותן נרצה לחפש במטריצה.
2. שם קובץ קלט המכיל מטריצה של אותיות (תפזורת, ראו תיאור בהמשך).
3. שם קובץ פלט אליו נכתוב את המילים שנמצאו וכן את מספר הפעמים שהופיעה כל מילה שנמצאה.
4. רצף אותיות המכיל כיווני חיפוש אשר יגדירו לנו את אופן החיפוש במטריצה.

את התכניות נפעיל ע"י קריאה לסקריפט משורת הפקודה.

דגשים לתרגיל:

1. יש לכתוב את כל התוכנית בקובץ אחד בשם **wordsearch.py**.
2. בהמשך יפורטו חלק מהפונקציות אותן תצטרכו לממש, אך מומלץ להתייחס אליהן בתור רשימה חלקית בלבד וכדאי מאוד לחלק את המשימות שפונקציות אלו אמורות לבצע למספר פונקציות קטנות יותר. את החלוקה הפנימית לפונקציות עליכם לבצע על פי שיקול דעתכם, לפי העקרונות שנלמדו בקורס. הדגש צריך להיות על קוד מודולרי (ללא כפל קוד), ברור וקריא.
3. ניקוד יינתן גם על יעילות הפתרונות. נסו לוודא שאינכם מבצעים בדיקות מיותרות.
4. שימו לב שהרצה מוצלחת של התכנית לא אמורה לגרור הדפסה כלשהי.
5. סגנון: הקפידו על תיעוד נאות ובחרו שמות משתנים משמעותיים. הקפידו להשתמש בקבועים (שמות משתנים באותיות גדולות) על פי הצורך.
6. אין להשתמש בחבילה **numpy** בתרגיל זה.
7. שימו לב - אומנם התכנית רצה משורת הפקודה, אבל את הפונקציות שאנחנו דורשים צריך ליישם בדיוק לפי הפירוט בהמשך.

תיאור הבעיה

עבור מטריצה של אותיות, למשל המטריצה הבאה:

	0	1	2	3	4
0	a	p	p	l	e
1	a	g	o	d	o
2	n	n	e	r	t
3	g	a	t	a	c
4	m	i	c	s	r

נרצה לדעת אילו מילים מתוך רשימת המילים נמצאות במטריצה, וכמה פעמים הן הופיעו.

רשימת מילים לדוגמה:

ort, can, toe, poet, cropdog, cat, ants, apple, cake, long, sh.

אנחנו מקבלים כקלט גם את כיווני החיפוש. נשים לב כי ישנם שמונה כיוונים לחיפוש במטריצה דו-מימדית:

כיוון	סימון (אות)	חץ
למעלה (לאורך אותה עמודה, מהשורה האחרונה עד לשורה הראשונה)	u	↑
למטה (לאורך אותה עמודה, מהשורה הראשונה עד לשורה האחרונה)	d	↓
ימינה (לאורך אותה שורה, מהעמודה הראשונה לעמודה האחרונה)	r	→
שמאלה (לאורך אותה שורה, מהעמודה האחרונה לעמודה הראשונה)	l	←
אלכסון עולה ימינה	w	↗
אלכסון עולה שמאלה	x	↖
אלכסון יורד ימינה	y	↘
אלכסון יורד שמאלה	z	↙

התכנית תבצע את החיפוש במטריצה על-פי האות הניתנת כפרמטר בשורת הפקודה. התכנית יכולה לקבל גם מספר אותיות (למשל lxw זה קלט תקין כפרמטר) ללא חשיבות לסדר, ובמקרה כזה החיפוש יתבצע בכל הכיוונים שניתנו כקלט.

עבור המטריצה ורשימת המילים שניתנו בדוגמא שלעיל, נראה דוגמאות לפלט בהינתן כיוונים שונים:

עבור u:

toe,1

עבור d:

poet,1

עבור l:

cat,1

dog,1

עבור w:

cat,1

עבור wl או lw:

cat,2

dog,1

בנספח מצורפות דוגמאות מפורטות עבור כלל כיווני החיפוש.

את התכנית נפעיל משורת הפקודה בלינוקס על ידי השורה הבאה:

python3 wordsearch.py word_file matrix_file output_file directions

<u>פרמטרים:</u>

1. **word_file**, שם קובץ המילים. קובץ המילים יכיל רשימה של מילים.
   - כל מילה מופיעה בשורה נפרדת.
   - ראו קובץ מילים לדוגמה - **word_list.txt**.
2. **matrix_file**, שם קובץ המטריצה (או קובץ התפזורת). קובץ המטריצה יכיל מטריצה של אותיות על פי הקידוד הבא:
   - כל שורה במטריצה נמצאת בשורה נפרדת בקובץ.
   - האותיות בכל שורה מופרדות ע"י פסיק.
   - מצורף קובץ **mat.txt** עם קידוד של המטריצה לדוגמה.
3. **output_file**, שם קובץ הפלט. אם הקובץ לא קיים, נפתח קובץ חדש בשם זה ואליו יוכנס פלט התכנית. אם קיים קובץ בשם זה נדרוס את התוכן שלו עם פלט התכנית.
4. **directions**, מחרוזת כיווני חיפוש . מחזרות המורכבת מרצף של אותיות המייצג את כיווני החיפוש המבוקשים בהתאם לטבלה לעיל.

הנחות על הקלט:

1. לא ניתן להניח שמילה תופיע רק פעם אחת באותו כיוון חיפוש, או בפרט פעם אחת באותה שורה/עמודה/אלכסון.
2. לא ניתן להניח כי אין חפיפה בין 2 מילים במטריצה, ובפרט אם זו אותה המילה. דוגמאות:

- ברשימת מילים מופיעות המילים dog, god, במטריצה מופיעה המילה dog משמאל לימין וכיווני החיפוש הם rl – במקרה זה על התוכנית שלכם למצוא גם את המילה dog וגם את המילה god.
- ברשימת המילים מופיעה המילה bob, במטריצה מופיעה המילה bobob מלמטה למעלה וכיוון החיפוש הוא u. במקרה זה על התוכנית שלכם למצוא את המילה bob פעמיים. אם כיווני החיפוש היו ud, התוכנית הייתה צריכה למצוא את המילה bob 4 פעמים בסך הכל.
- ברשימת מילים מופיעות המילים bob, cat, bobcat, במטריצה מופיעה המילה bobcat משמאל לימין ומוגדרים כיווני חיפוש rl. במקרה זה על התוכנית שלכם למצוא את 3 המילים הנ"ל (ובפרט למצוא את המילה bob פעמיים סך-הכל כבדוגמא הקודמת).
- ברשימת מילים מופיעה המילה red ובמטריצה מופיעה המילה dered מלמעלה למטה ומוגדרים כיווני החיפוש ud, על התוכנית שלכם למצוא את המילה red פעמיים סך הכל.

3. לא ניתן להניח שהמטריצות ריבועיות.
4. ניתן להניח שקבצי הקלט, אם קיימים, נמצאים בפורמט המתואר ובעלי הרשאות קריאה. בפרט, לא מופיעה בהם שורה ריקה מלבד השורה האחרונה.
5. ניתן להניח שכל מילה בקובץ המילים מופיעה רק פעם אחת ברשימה זו (במטריצה היא יכולה להופיע יותר מפעם אחת, כאמור לעיל).
6. ניתן להניח שהמטריצות מלבניות (כלומר כל אורכי השורות במטריצה שווים).
7. ניתן להניח שלא תוזן מחרוזת ריקה בתור הקלט למחרוזת הכיוונים.

פלט:

הפלט של החיפוש הוא רשימה של המילים מקובץ המילים שנמצאו במטריצת הקלט ומספר ההופעות של כל מילה. בכל שורה יופיע צמד אחד בלבד - מילה ומספר ההופעות שלה, כשהם מופרדים ביניהם בפסיק. הפלט כולל כמובן רק מילים שמופיעות במטריצה לפחות פעם אחת. לתרגיל מצורפות מספר דוגמאות לקבצי פלט לדוגמה לשימושכם.

שימו לב: שורה היא מחרוזת, ללא התו "\n", שלאחר המחרוזת יש תו "\n". הגדרה זו נכונה גם עבור השורה האחרונה בקובץ, וכן נכונה גם בקבצי הקלט וגם בקבצי הפלט.

<u>טיפול בקלט:</u>

במידה ומגיע קלט לא תקין לפי המצבים הבאים המתוארים, יש להדפיס הודעת שגיאה אינפורמטיבית אשר תשקף את הבעיה, ותנוסח לפי שיקולכם, ולאחר מכן לסיים את הריצה של התוכנית:

1. במקרה שמספר הפרמטרים לא תקין (שונה מ-4).
2. במקרה שקובץ המילים או קובץ המטריצה לא קיים. אם שניהם לא קיימים מספיק להדפיס את ההודעה עבור קובץ המילים בלבד.
3. במקרה שקלט כיווני החיפוש מכיל כיווני חיפוש לא חוקיים (על פי הטבלה).

במידה ויש יותר מבעיה אחת בקלט מספיק לציין אחת מביניהן.

שימו לב:

- אין משמעות להופעה של אותו כיוון חיפוש יותר מפעם אחת והתכנה אמורה להתעלם מכך (יש להתייחס למקרה כזה כאילו אותו כיוון חיפוש הופיע פעם אחת בלבד).
- במקרה שלא נמצאה אף מילה, קובץ הפלט צריך להיות ריק.
- קבצים ריקים הם גם קלט תקין - אם המטריצה ריקה (כלומר בעלת 0 שורות) או שקובץ המילים ריק, אזי גם קובץ הפלט יהיה ריק.
- case sensitivity - מילים יכולות להופיע ברשימת מילים ובמטריצה באותיות לועזיות גדולות או קטנות, ובמקרה כזה יש להתייחס אליהן כאל מילים נפרדות. לדוגמה, ברשימת המילים מופיעות המילים cat, Cat ו-CAT ואלו נחשבות למילים שונות. אם מבין מילים רק המילה cat מופיעה במטריצה בכיוון בו מבצעים חיפוש, הרי שאין להתייחס למילים Cat ו-CAT כאילו נמצאו, ובקובץ הפלט תופיע רק המילה cat.
- כיווני החיפוש התקינים מפורטים בטבלה בעמ' 2, ובאותיות קטנות בלבד.

פונקציות למימוש

1. עליכם לממש את הפונקציה **read_wordlist(filename)**.

- הפונקציה מקבלת פרמטר אחד – **filename**, שם קובץ הקלט המכיל את רשימת המילים ומיוצג בתור מחרוזת.
- על הפונקציה לפתוח את הקובץ ששמו ניתן כפרמטר, לקרוא את המילים שבתוכו הכתובות בפורמט שתואר לעיל ולהחזיר רשימה של המילים שנקראו, **לפי הסדר**.

2. עליכם לממש את הפונקציה **read_matrix(filename)**.

- הפונקציה מקבלת פרמטר אחד – **filename**, שם קובץ הקלט המכיל את מטריצת האותיות ומיוצג בתור מחרוזת.
- על הפונקציה לפתוח את הקובץ ששמו ניתן כפרמטר, לקרוא את מטריצת האותיות שבתוכו הכתובה בפורמט שתואר לעיל ולהחזיר רשימה דו מימדית של אותיות המטריצה (רשימה אחת שאיבריה הן רשימות, כל רשימה מייצגת רשימה אחת, שאיבריהן הן רשימה של אותיות).

3. עליכם לממש את הפונקציה **find_words(word_list, matrix, directions)**.

- הפונקציה מקבלת 3 פרמטרים:
  - **word_list**, רשימת המילים לחיפוש.
  - **matrix**, רשימה דו מימדית המייצגת את מטריצת האותיות.
  - **directions**, מחרוזת אותיות המייצגות את הכיוונים לחיפוש במטריצה, כפי שתואר לעיל.
- הפונקציה תחפש כל אחת מהמילים ברשימת המילים בתור מטריצת האותיות, לפי כיווני החיפוש שהתקבלו.
- **סדר התוצאות שנמצאות - אינו חשוב.**
- על הפונקציה להחזיר רשימה של זוגות (כלומר איברי הרשימה מסוג tuple שכל אחד מהם מכיל שני איברים), כל זוג מהצורה **(word, count)** כאשר **word** הוא מילה (מטיפוס str) המופיעה במטריצה, ו-**count** הוא כמות הפעמים שאותה מילה נמצאה במטריצה (מטיפוס int).

שימו לב: ההמלצה על חלוקת הפונקציות למימוש לפונקציות פנימיות מתייחסת בעיקר לפונקציה זו (אך ממש לא רק בהכרח). הפונקציה הזו מבצעת חלק ניכר מפעולת התוכנית, שניתן לחלק לכמה משימות משניות, ומומלץ לבצע כל משימה משנית בפונקציה נפרדת. החלוקה למשימות נפרדות היא לפי שיקול דעתכם.

4. עליכם לממש את הפונקציה **write_output(results, filename)**.
   - הפונקציה מקבלת 2 פרמטרים:
     - **results**, רשימה של זוגות, כל זוג מהצורה **(word, count)**, כפי שהתקבלה כערך החזרה מהפונקציה find_words.
     - **filename**, שם קובץ הפלט, אליו יכתבו תוצאות החיפוש.
   - הפונקציה תייצר את הקובץ (או תדרוס אם כבר קיים) ששמו ניתן כפרמטר, ותכתוב לתוכו את תוצאות החיפוש של המילים במטריצת האותיות, בפורמט שהוגדר לעיל.
   - יש לממש את הפונקציה בצורה כזו כך **שסדר המילים בקובץ הפלט יהיה כסדר המילים שהועברו לפונקציה**

5. עליכם לממש פונקציה ראשית בשם שתבחרו, שמבצעת קריאות לכל הפונקציות האחרות שהוזכרו ופונקציות שלכם (בצורה ישרה או עקיפה), ותחבר את כל חלקי התרגיל יחד. הבחירה של הפרמטרים וערכי ההחזרה לשיקולכם. לפונקציה זו עליכם לקרוא בסוף הקובץ באזור ה-main, כלומר בהזחה מתחת לשורה:

```
if __name__ == "__main__":
    # your code here
```

בדיקת התוכנית

אנו ממליצים לכתוב את התוכנית בשלבים. לאחר כל פונקציה שתכתבו, הקפידו לבדוק אותה ע"י קריאה לתוכנית עם פרמטרים שונים והשוואת התוצאות שקיבלתם למה שמצופה. חשבו על קלטים שונים לפונקציות שיכולים לגרום לתוצאות שונות, כולל מקרי קצה. אנו מאוד ממליצים שלא לכתוב את כל התוכנית יחד ולבדוק אותה בשלמותה בסוף, מכיוון שהדבר יקשה עליכם מאוד במציאה וטיפול בבעיות.

הוראות הגשה

עליכם להגיש קובץ בשם **ex5.zip** בקישור ההגשה של תרגיל 5 דרך אתר הקורס על ידי לחיצה על "Upload file". הקובץ **ex5.zip** צריך להכיל אך ורק את הקובץ **wordsearch.py**

הנחיות כלליות בנוגע להגשה

- הנכם רשאים להגיש תרגילים דרך מערכת ההגשות באתר הקורס מספר רב של פעמים. ההגשה האחרונה בלבד היא זו שקובעת ושתיבדק.
- לאחר הגשת התרגיל, ניתן ומומלץ להוריד את התרגיל המוגש ולוודא כי הקבצים המוגשים הם אלו שהתכוונתם להגיש וכי הקוד עובד על פי ציפיותיכם.
    - באחריותכם לוודא כי – PDF הבדיקות נראה כמו שצריך.
- קראו היטב את קובץ נהלי הקורס לגבי הנחיות נוספות להגשת התרגילים.
- שימו לב - יש להגיש את התרגילים בזמן!

בהצלחה!

*** מצורף נספח בדף הבא**

נספח

עבור הטבלה:

a	p	p	l	e
a	g	o	d	o
n	n	e	r	t
g	a	t	a	c
m	i	c	s	r

ורשימת המילים:

long, short, can, toe, poet, crop, dog, cat, ants, apple, cake

הפלט עבור כלל כיווני החיפוש:

u:

toe,1

a	p	p	l	e
a	g	o	d	o
n	n	e	r	t
g	a	t	a	c
m	i	c	s	r

d:

poet,1

a	p	p	l	e
a	g	o	d	o
n	n	e	r	t
g	a	t	a	c
m	i	c	s	r

r:

apple,1

a	p	p	l	e
a	g	o	d	o
n	n	e	r	t
g	a	t	a	c
m	i	c	s	r

l:

cat,1

dog,1

a	p	p	l	e
a	g	o	d	o
n	n	e	r	t
g	a	t	a	c
m	i	c	s	r

w:

cat,1

a	p	p	l	e
a	g	o	d	o
n	n	e	r	t
g	a	t	a	c
m	i	c	s	r

x:

can,1

crop,1

a	p	p	l	e
a	g	o	d	o
n	n	e	r	t
g	a	t	a	c
m	i	c	s	r

y:

ants,1

a	p	p	l	e
a	g	o	d	o
n	n	e	r	t
g	a	t	a	c
m	i	c	s	r

z:

long,1

a	p	p	l	e
a	g	o	d	o
n	n	e	r	t
g	a	t	a	c
m	i	c	s	r