

# בית הספר להנדסה ומדעי המחשב ע"ש רחל וסלים בנין

מבוא למדעי המחשב 67101

## תרגיל 5 - רשימות ושימוש בקבצים

להגשה בתאריך 21/11/2018 בשעה 22:00

### הקדמה

בתרגיל זה נתרגל שימוש בלולאות, רשימות רב מימדיות, מילונים, פרמטרים בשורת פקודה ועבודה עם קבצים.

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 תקינות 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 ומספר הפעמים שהופיעו.
  4. רצף אותיות המכיל כיווני חיפוש אשר יגדירו לנו את אופן החיפוש במטריצה.
- את התכניות נפעיל ע"י קריאה לסקריפט משורת הפקודה.

### דגשים לתרגיל:

- יש לכתוב את כל התוכנית בקובץ אחד ושמו wordsearch.py. 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. את החלוקה לפונקציות יש לנמק בקובץ ה-README.
- כל הקבצים יוגשו בתוך הקובץ ex5.zip.
- ניקוד יינתן גם על יעילות הפתרונות. נסו לוודא שאינכם מבצעים בדיקות מיותרות.
- שימו לב שהרצה מוצלחת של התכנית לא אמורה לגרור הדפסה כלשהי.
- סגנון: הקפידו על תיעוד נאות ובחרו שמות משתנים משמעותיים. הקפידו להשתמש בקבועים (שמות משתנים באותיות גדולות) על פי הצורך.

בית הספר להנדסה ומדעי המחשב ע"ש רחל וסלים בנין

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:

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

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

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

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

## בית הספר להנדסה ומדעי המחשב ע"ש רחל וסלים בנין

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
```

### פרמטרים:

- **word\_file**: שם קובץ המילים - יכיל רשימה של מילים. כל מילה נמצאת בשורה נפרדת. הקובץ מסתיים בשורה ריקה (כלומר המילה האחרונה בקובץ מסתיימת גם היא בתו ירידת שורה '\n'). ראו קובץ מילים לדוגמה - word\_list.txt.
- **matrix\_file**: שם קובץ המטריצה (או קובץ התפזורת) - יכיל מטריצה של אותיות על פי הקידוד הבא: כל שורה במטריצה נמצאת בשורה נפרדת בקובץ, האותיות בשורה מופרדות ע"י פסיק. הקובץ מסתיים בשורה ריקה (כלומר השורה האחרונה של המטריצה בקובץ מסתיימת גם היא בתו ירידת שורה '\n'). מצורף קובץ mat.txt עם קידוד של המטריצה לדוגמה.
- **output\_file**: שם קובץ הפלט - אם הקובץ לא קיים, נפתח קובץ חדש בשם זה במיקום בו הוא ניתן (ואם לא ניתן מיקום, הקובץ יפתח בתיקייה הנוכחית), ואליו יוכנס פלט התכנית. אם קיים קובץ בשם זה במיקום בו הוא ניתן (ואם לא ניתן מיקום, אז נבדוק אם הוא קיים בתיקייה הנוכחית), נדרוס את התוכן שלו עם פלט התכנית.
- **directions**: כיווני חיפוש - רצף של אותיות המייצג את כיווני החיפוש בהתאם לטבלה לעיל.

## בית הספר להנדסה ומדעי המחשב ע"ש רחל וסלים בנין

### הנחות על הקלט:

- לא ניתן להניח שמילה תופיע רק פעם אחת באותו כיוון חיפוש, או בפרט פעם אחת באותה שורה/עמודה/אלבסון.
- לא ניתן להניח כי אין חפיפה בין 2 מילים במטריצה, ובפרט אם זו אותה המילה. דוגמאות:
  - אם ברשימת מילים מופיעות המילים dog, god, ובמטריצה מופיעה המילה dog רשומה בשורה מסויימת משמאל לימין, וכן כיווני החיפוש הם rl, על התוכנית שלכם למצוא גם את המילה dog וגם את המילה god.
  - אם ברשימת מילים מופיעות המילים bobcat, cat, bob, ובמטריצה מופיעה המילה bobcat משמאל לימין, ומוגדרים גם כיווני חיפוש rl, על התוכנית שלכם למצוא את 3 המילים הנ"ל (ובפרט למצוא את המילה bob פעמיים סך-הכל).
  - אם 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.
  - אם ברשימת המילים מופיעה המילה bob ובמטריצה מופיעה המילה bobob מלמעלה למטה ומוגדר כיוון החיפוש u, על התוכנית שלכם למצוא את המילה bob פעמיים.
- לא ניתן להניח שהמטריצות ריבועיות.
- ניתן להניח שקבצי הקלט, אם קיימים, נמצאים בפורמט המתואר ובעלי הרשאות קריאה.
- ניתן להניח שכל מילה בקובץ המילים מופיעה רק פעם אחת.
- ניתן להניח שהמטריצות מלבניות (כלומר כל אורכי השורות במטריצה שווים).

### פלט:

הפלט של החיפוש הוא רשימה של מילים מרשימת המילים ומספר הפעמים שמילים אלו הופיעו במטריצת הקלט (המילה והמספר מופרדים בפסיק). כל צמד של מילה ומספר ההופעות שלה נמצא בשורה נפרדת. הפלט כולל רק מילים שמספר ההופעות שלהן במטריצת הקלט הוא לפחות 1. לתרגיל מצורפות מספר דוגמאות לקבצי פלט לדוגמה לשימושכם.

### טיפול בקלט:

- במידה ומגיע קלט לא תקין לפי המצבים הבאים המתוארים, יש להדפיס הודעת שגיאה אינפורמטיבית אשר תשקף את הבעיה, ותנסח לפי שיקולכם, ולאחר מכן לסיים את הריצה של התוכנית:
- במקרה שמספר הפרמטרים לא תקין (שונה מ-4)
  - במקרה שקובץ המילים או קובץ המטריצה לא קיים. אם שניהם לא קיימים מספיק להדפיס את ההודעה עבור קובץ המילים בלבד.
  - במקרה שקלט כיווני החיפוש מכיל כיווני חיפוש לא חוקיים (על פי הטבלה).

## בית הספר להנדסה ומדעי המחשב ע"ש רחל וסלים בנין

### שימו לב:

- אין משמעות להופעה של אותו כיוון מספר פעמים והתכנה אמורה להתעלם מכך (נחשב 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, קובץ הפלט יהיה ריק.
- מילים יכולות להיות רשומות ברשימת מילים ובמטריצה באותיות לועזיות גדולות וקטנות, וזה נחשב למילים נפרדות. לדוגמה, יכולות להיות רשומות ברשימת מילים המילים `cat`, `Cat` ו-`CAT` ואלו נחשבות למילים שונות, ואם במטריצה רשומה המילה `cat` בלבד בכיוון כלשהו שבו גם מבצעים חיפוש, רק המילה `cat` מרשימת המילים אמורה להמצא ולהרשם בקובץ הפלט.

### פונקציות למימוש

- עליכם לממש את הפונקציה `check_input_args(args)`. הפונקציה מקבלת כקלט פרמטר אחד:
  - `args`: רשימת הארגומנטים (מחרוזות) שהתוכנית קיבלה משורת הפקודה.על הפונקציה לבדוק כל אחד מהפרמטרים, כפי שתואר לעיל. במידה ונמצאה שגיאה באחד הפרמטרים, על הפונקציה להחזיר את הודעת השגיאה המתאימה (אבל אינה מדפיסה אותה). במידה ולא נמצאו שגיאות כלל, על הפונקציה להחזיר את הערך `None`.
- עליכם לממש את הפונקציה `read_wordlist_file(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.
- עליכם לממש את הפונקציה `read_matrix_file(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 שורה של אותיות במטריצה).
- עליכם לממש את הפונקציה `find_words_in_matrix(word_list, matrix, directions)`. הפונקציה מקבלת 3 פרמטרים:
  - `word_list`: רשימה המילים לחיפוש.
  - `matrix`: רשימה דו מימדית המייצגת את מטריצת האותיות.
  - `directions`: מחרוזת אותיות המייצגות את הכיוונים לחיפוש במטריצה, כפי שתואר לעיל.

## בית הספר להנדסה ומדעי המחשב ע"ש רחל וסלים בנין

הפונקציה תחפש כל אחת מהמילים ברשימת המילים בתור מטריצת האותיות, לפי כיווני החיפוש שהתקבלו. הפונקציה תשמור את מספר ההופעות לכל מילה שנמצאה במבנה נתונים מיוחד שבחירתו נתונה לשיקולכם. 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) 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, וכל משימה משנית לבצע בפונקציה נפרדת. החלוקה למשימות נפרדות היא לפי שיקול דעתכם.

- עליכם לממש את הפונקציה `write_output_file(results, output_filename)`. הפונקציה מקבלת 2 פרמטרים:
  - **results:** רשימה של זוגות, כל זוג מהצורה (word, count), כפי שהתקבלה כערך החזרה מהפונקציה `find_words_in_matrix`.
  - **output\_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.
- עליכם לממש פונקציה ראשית בשם שתבחרו, שמבצעת קריאות לכל הפונקציות האחרות שהוזכרו ופונקציות שלכם (בצורה ישירה או עקיפה), ותחבר את כל חלקי המשחק יחדיו. 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 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.

בנוסף, עליכם לכתוב פונקציה אשר בודקת את אחת הפונקציות בתוכנית שלכם לפי ההנחיות הבאות:

- עליכם לבדוק את אחת הפונקציות שמימשתם אשר נקראת מתוך הפונקציה `find_words_in_matrix`, כאשר את הפונקציה הספציפית שתבדקו אתם תבחרו בעצמכם.
- את הפונקציה יש לממש בקובץ נפרד בשם `check_wordsearch.py`.
- השם וחתימת פונקציית הבדיקה נתונים לבחירתכם.
- בחרו פונקציה אשר מקבלת לפחות פרמטר אחד, וכן מחזירה לפחות ערך החזרה אחד.

## בית הספר להנדסה ומדעי המחשב ע"ש רחל וסלים בנין

- על הפונקציה לקרוא לפונקציה הפנימית שבחרתם בעזרת קלטים שונים כדי לבדוק מצבים שונים איתם הפונקציה מתמודדת, לבחירתכם. הקפידו לחשוב ולבחור מקרים סטנדרטיים, וכן מקרי קצה.
- ודאו שערכי ההחזרה שהתקבלו מהפונקציה זהים לאלו שמצופה מהם להיות.
- קראו לפונקציית הבדיקה בקובץ הבדיקה באזור ה-main, כלומר בהזחה מתחת לשורה:  
`if __name__ == "__main__":`  
`# your code here`
- **כתבו בקובץ ה-README את שם הפונקציה אותה בחרתם לבדוק, את הקלטים שאיתם בחרתם לבדוק אותה ומדוע בחרתם לבדוק מקרים אלו.**

### הוראות הגשה

- עליכם להגיש את הקובץ ex5.zip (בלבד) בקישור ההגשה של תרגיל 5 דרך אתר הקורס על ידי לחיצה על "Upload file". אנו ממליצים להתחיל לעבוד על התרגיל בשלב מוקדם שכן התרגיל ארוך מקודמיו.
- ex5.zip צריך לכלול (לפחות) את הקבצים:

1. wordsearch.py
2. check\_wordsearch.py
3. README - כולל את החלקים המפורטים בנהלי הקורס, הסברים על החלוקה הפנימית לפונקציות, וכן תיאור הבדיקות שביצעתם בפונקציית הבדיקה כפי שתואר לעיל.

### 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!

## בית הספר להנדסה ומדעי המחשב ע"ש רחל וסלים בנין

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



## בית הספר להנדסה ומדעי המחשב ע"ש רחל וסלים בנין

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

# בית הספר להנדסה ומדעי המחשב ע"ש רחל וסלים בנין

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