תרגיל בית 4

1. יש לכתוב תכנית בקובץ matrix.py העוסקת בפעולות על מטריצות לפי ההנחיות הבאות:  
   (בתרגיל זה המלה "מטריצה" מתייחסת לרשימה של רשימות מספרים שכולן באותו גודל)
   1. יש לכתוב פונקציה המקבלת מטריצה A ומחזירה את המטריצה המשוחלפת שלה AT (המטריצה שהשורות שלה הן העמודות של A). יש לכתוב שתי גירסאות לפונקציה: אחת עם לולאות ואחת הכוללת להכיל פקודה אחת בלבד ללא לולאות על ידי שימוש ב list comprehension
   2. יש לכתוב פונקציה add\_matrix שמקבלת שתי מטריצות A ו B ומחזירה מטריצה שבה כל איבר הוא סכום האיברים המקבילים במטריצות A וB.
   3. יש לכתוב פונקציה print\_matrix שמקבלת מטריצה ומדפיסה אותה כך שכל שורה במטריצה מופיעה בשורה נפרדת בפלט.

התכנית הראשית תקרא מקובץ קלט matrices.txt שתי מטריצות A ו B ומדפיסה את תוצאת החיבור AT + B. כל שורה במטריצה נמצאת בשורה נפרדת בקובץ. אפשר להניח שמספר העמודות ב A זהה למספר השורות ב B.

דוגמה לקובץ קלט:

A=

1 2 3

4 5 6

B=

1 2

3 4

5 6

במקרה זה יודפס:

2 6

5 9

8 12

1. **צופן הצבה** הוא שיטת הצפנה לטקסט, שבה כל אות בא"ב האנגלי מוחלפת באות אחרת בא"ב האנגלי. כלומר, מפתח ההצפנה הוא תמורה על 26 האותיות. למשל, מפתח הצפנה הבא

groexuwtdmzvbjqkfnchliysap

אומר שהאות a מוצפנת באמצעות g, האות b מוצפנת באמצעות r, האות c מוצפנת באמצעות o וכן הלאה.

יש לכתוב תכנית בקובץ crypto.py שמציגה למשתמש תפריט ממנו הוא יכול לבחור את אחת הפעולות הבאות:

1. יצירת מפתח: אם המשתמש הקיש k (קיצור של key), התוכנית תייצר מפתח הצפנה **אקראי** ותכתוב אותו לתוך קובץ key.txt. יש לכתוב לקובץ את המפתח בצורת מחרוזת כמו שמופיע בדוגמה (groexuwtdmzvbjqkfnchliysap).
2. הצפנה: אם המשתמש הקיש e (קיצור של encrypt), התוכנית תצפין את הטקסט שנמצא בקובץ plaintext.txt באמצעות המפתח שנמצא ב key.txt. התוצאה תודפס לקובץ בשם ciphertext.txt. התוכנית תצפין רק אותיות אנגליות ותתעלם מכל תו שאינו אות אנגלית. אותיות גדולות יוצפנו כאותיות קטנות.
3. פענוח: אם המשתמש הקיש d (קיצור של decrypt), התוכנית תקרא מפתח הצפנה מתוך קובץ key.txt ותפענח את הטקסט שנמצא בתוך קובץ בשם ciphertext.txt. התוצאה תיכתב לתוך קובץ decrypted.txt.
4. סיום: אם המשתמש הקיש כל דבר אחר, התכנית מסתיימת ללא פעולה נוספת

התוכנית תשתמש במילון כדי לשמור את מפתח ההצפנה. כל זוג במילון יכיל אות ואת האות שמצפינה אותה. למשל, המפתח בדוגמה למעלה יישמר במילון

key = {'a':'g', 'b': 'r', 'c': 'o', ...}

שימו לב שכדי להריץ את התכנית, יש להכין קודם קובץ plaintext.txt שמכיל טקסט כלשהו

1. יש לכתוב תכנית minesweeper.py המאפשרת למשתמש לשחק במשחק "שולה המוקשים". למי שלא מכיר: [הנה הסבר מפורט](http://he.wikipedia.org/wiki/%D7%A9%D7%95%D7%9C%D7%94_%D7%94%D7%9E%D7%95%D7%A7%D7%A9%D7%99%D7%9D_%28%D7%9E%D7%A9%D7%97%D7%A7%29) מויקיפדיה וכן מומלץ לשחק קצת ב"זמנכם החופשי": <http://minesweeperonline.com>

הדגשים בדרישות התרגיל הם: חלוקה נכונה לפונקציות, עבודה עם רשימות מקוננות ולולאות מקוננות, רקורסיה, וכן ידידותיות מול המשתמש.

הכוונה היא לממש את המשחק בצורה פשוטה. אין צורך לממש את כל האפשרויות המתקדמות של שולה מוקשים "אמיתי".

התוכנית תבצע את הפעולות הבאות

* תבקש מהמשתמש לבחור את גודל הלוח ואת מספר המוקשים. (לשם הפשטות - לוח ריבועי בגודל 9x9 לכל היותר. מספר המוקשים לא יותר מפעמיים גודל הלוח. כלומר, בלוח nxn, לא יותר מ 2n מוקשים)
* תפזר את המוקשים באופן אקראי.
* בכל מהלך יש לקבל מהמשתמש את הקואורדינטות של המשבצת לחשיפה.
* התכנית תדפיס את הלוח לאחר כל מהלך. כאשר בכל משבצת שנחשפה יש להדפיס את תוכנה (אם אין בה מוקש) – כלומר את מספר המוקשים במשבצות השכנות.
* תממש את המקרה של חשיפת משבצת שאינה בסמיכות לשום מוקש. במקרה זה יחשפו כל המשבצות השכנות ברקורסיה עד למציאת משבצות שיש בשכנותן מוקשים. **(יש לשחק קצת ולצבור ניסיון במשחק לפני תחילת הקידוד...)**
* אין צורך לממש את האפשרות לסימון מוקשים (כפי שאפשר לעשות במשחק המקורי בלחיצת עכבר ימנית)
* תזהה מתי מסתיים המשחק: השחקו מפסיד אם הוא בוחר משבצת שיש בה מוקש. השחקן מנצח אם הוא חשף את כל המשבצות שאין בהן מוקש.
* תצוגה: משבצת ריקה – טרם נחשפה. ספרה במשבצת – מספר המוקשים השכנים. X במשבת – מוקש.

**תצוגה בסוף משחק:**

+---+---+---+---+---+

1| x | x | 1 | | |

+---+---+---+---+---+

2| 2 | 2 | | | x |

+---+---+---+---+---+

3| 0 | 1 | | 4 | x |

+---+---+---+---+---+

4| 0 | 1 | x | x | |

+---+---+---+---+---+

5| 0 | 1 | | | |

+---+---+---+---+---+

1 2 3 4 5

**תצוגה אחרי 3 מהלכים:**

+---+---+---+---+---+

1| | | 1 | | |

+---+---+---+---+---+

2| 2 | 2 | | | |

+---+---+---+---+---+

3| 0 | 1 | | 4 | |

+---+---+---+---+---+

4| 0 | 1 | | | |

+---+---+---+---+---+

5| 0 | 1 | | | |

+---+---+---+---+---+

1 2 3 4 5

**תצוגה אחרי מהלך אחד:**

+---+---+---+---+---+

1| | | | | |

+---+---+---+---+---+

2| | | | | |

+---+---+---+---+---+

3| | | | 4 | |

+---+---+---+---+---+

4| | | | | |

+---+---+---+---+---+

5| | | | | |

+---+---+---+---+---+

1 2 3 4 5

**תצוגה ראשונית:**

+---+---+---+---+---+

1| | | | | |

+---+---+---+---+---+

2| | | | | |

+---+---+---+---+---+

3| | | | | |

+---+---+---+---+---+

4| | | | | |

+---+---+---+---+---+

5| | | | | |

+---+---+---+---+---+

1 2 3 4 5

הנחיות הגשה:

1. אין להשתמש במודולים מלבד מודולים סטנדרטיים כמו math, random, sys, timeit
2. יש לפתור כל שאלה בקובץ נפרד עם סיומת .py
3. יש להגיש את כל הקבצים בקובץ אחד מכווץ עם סיומת .zip
4. כל קובץ יתחיל בהערה ובה המידע הבא:
   1. שם הסטודנט
   2. מס' תעודת זהות
   3. מספר דף התרגילים
   4. שם התוכנית

למשל, עבור תרגיל 1 בדף 2:

"""

Student: Wolfgang Amadeus Mozart

ID: 121212121

Assignment no. 5

Program: Miller\_Rabin.py

"""

שימו לב! יש להקפיד על הנחיות ההגשה האלה. הגשה שלא בדיוק בפורמט הזה לא תקבל את מלוא הנקודות ואף עלולה להיפסל.