תרגיל בית 6

1. בתרגיל זה נכתוב תכנית minesweeper.py המאפשרת למשתמש לשחק ב "שולה המוקשים". למי שלא מכיר: הנה הסבר מפורט 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, הגדרת מחלקה 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 4x4 לפחות ו 9x9 לכל היותר. מספר המוקשים לא יותר מפעמיים גודל הלוח. כלומר, בלוח nxn, לא יותר מ 2n מוקשים)
- תפזר את המוקשים באופן אקראי על הלוח.
- בכל מהלך התוכנית תקבל מהמשתמש את הקורדינטות של המשבצת לחשיפה.
- 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: השחקן מפסיד אם הוא בוחר משבצת שיש בה מוקש. השחקן מנצח אם הוא חשף את כל המשבצות שאין בהן מוקש.
- תצוגה: משבצת ריקה – טרם נחשפה. סיפרה במשבצת – מספר המוקשים השכנים. X במשבת – מוקש.

יש לכתוב מחלקה MSSquare (קיצור של minesweeper square) שמגדירה משבצת בתוח. לכל אובייקט של המחלקה יהיו שלוש תכונות:

- has_mine – משתנה סוליני שהוא True אם ורק אם במשבצת יש מוקש
- hidden – משתנה בוליאני שהוא True אם המשבצת עוד לא נחשפה
- neighbor_mines – משתנה int שמתאר את מספר המוקשים במשבצות השכנות (כולל משבצות ששכנות באלכסון.

יש לכתוב בנאי עבור המחלקה (__init__), להגדיר משתנים פנימיים ו getters ו setters (@property).

הלוח יהיה רשימה מקוננת של משבצות מטיפוס MSSquare

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
```

2. בתרגיל זה נשתמש במחלקה Point שאותה הגדרנו בהרצאה. עליכם לכתוב מחלקה Line המגדירה ישר במישור. הישר יוגדר באמצעות שתי תכונות מטיפוס Point : __p ו __q. הישר הוא הישר העובר בין שתי הנקודות.

א. שיטות במחלקה Point :

- הוסיפו ל Point שיטות getters ו setters. ב- setters יש לזרוק חריגה במקרה שהארגומנט אינו מספר.
- הוסיפו ל Point שיטה __init__. יש לזרוק חריגה במקרה שהארגומנט אינו מספר.
- הוסיפו ל Point שיטה __str__. יש להציג נקודות עם שתי ספרות אחרי הנקודה העשרונית.

ב. שיטות במחלקה Line :

- כתבו ל Line שיטה __init__ המקבלת כארגומנטים שתי נקודות
- כתבו ל Line שיטות getters ו setters. ב- setters יש לזרוק חריגה במקרה שהארגומנט אינו מטיפוס Point.
- כתבו ל Line שיטה is_vertical(self) המחזירה True אם הישר הוא אנכי
- כתבו ל Line שיטה slope(self) המחזירה את שפוע הישר. אם הישר אנכי יוחזר None
- כתבו ל Line שיטה y_intersect(self) המחזירה את נקודת החיתוך עם ציר ה y. (אם הישר אנכי יוחזר None).
- כתבו ל Line שיטה __str__ המחזירה מחרוזת שמתארת משוואת ישר מהצורה y = ax+b (עבור קו אנכי המשוואה תהיה מהצורה x = a. למשל, 'y = 2.05x + 1.95' או 'x = 2.0'.
- כתבו ל Line שיטה parallel(self, other) המחזירה True אם self מקביל ל other (כולל המקרה שבו הישרים זהים)
- כתבו ל Line שיטה equals(self, other) המחזירה True אם self ו other מתארים את אותו ישר. (שימו לב שהם עשויים לתאר את אותו הישר גם אם הנקודות __p ו __q שלהם שונות.)
- כתבו ל Line שיטה intersection(self, other) שמחזירה את נקודת החיתוך של self ו other. אם הישרים לא נחתכים יוחזר None.

ג. כתבו תוכנית lines.py שקוראת מקובץ input.txt נתונים של קווים. כל קו בשורה נפרדת, כך שכל קו מורכב מ-4 מספרים המתאימים לשתי נקודות. התוכנית תדפיס לקובץ output.txt את משוואות כל הישרים ולכל ישר את נקודות החיתוך עם הישרים הקודמים. אם ישרים הם מקבילים או זהים תודפס הודעה מתאימה.
התוכנית תדפיס Error במקרה שהקלט אינו מורכב ממספרים או שיש בשורה פחות מ 4 מספרים.

מצורפות דוגמאות קלט ופלט.

0111401 מבוא למדעי המחשב
החוג למדעי המחשב, המכללה האקדמית תל-חי
סמסטר סתיו תשפ"א

הנחיות הגשה:

1- יש להגיש תוכניות שרצות ללא שגיאות. תוכנית שתוגש עם שגיאות תקבל לכל היותר חצי מהנקודות.
2- יש לכתוב הערות לתוכנית: docstring בתחילת כל פונקציה, הסבר קצר בתחילת התוכנית, הסבר בתחילת לולאות.
3- אין להשתמש במודולים מלבד מודולים סטנדרטיים כמו math, random, sys,
4- יש לפתור כל שאלה בקובץ נפרד עם סיומת py.
5- יש להגיש את כל הקבצים בקובץ אחד מכווץ עם סיומת zip. שם קובץ ה zip צריך להיות מספר הת"ז שלכם ומספר עבודת הבית. למשל, 313131313_hw6.zip
6- כל קובץ יתחיל בהערה ובה המידע הבא:
   א. שם הסטודנט
   ב. מס' תעודת זהות
   ג. מספר דף התרגילים
   ד. שם התוכנית

למשל, עבור תרגיל 1 בדף 6:

```
"""
Student: Jennifer Lopez
ID: 313131313
Assignment no. 6
Program: minesweeper.py
"""
```

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