מטלת מנחה (ממ"ן) 14

הקורס: 20441 - מבוא למדעי המחשב ושפת Java

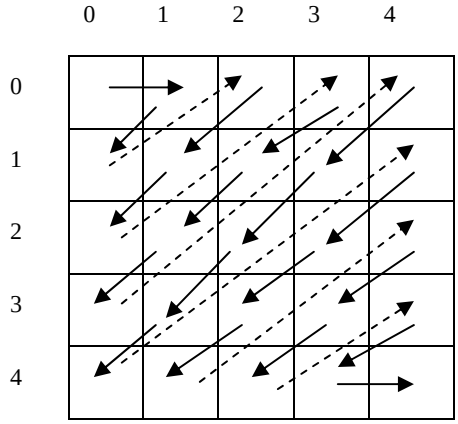
חומר הלימוד למטלה: יחידות 8,9,10,13 **נושא המטלה:** מערכים ולולאות

מספר השאלות: 2 **משקל המטלה:** 4 נקודות

סמסטר: 2009ב **מועד אחרון להגשה:** 16.5.2009

(ת)

נתון המערך הדו-מימדי הבא (**המערך לא בהכרח ריבועי**)



לכל תא במערך (מלבד התא הימני בשורה התחתונה) מוגדר תא עוקב על-פי **דפוס אלכסוני** בסדר שמופיע בתמונה. הדפוס מוגדר באופן הבא:

1. אם התא הנוכחי הוא בשורה התחתונה במערך, האיבר העוקב שלו הוא בראש האלכסון הבא מימין. למשל, בדוגמא למעלה, העוקב של תא (4, 1) הוא התא (2, 4).
2. אחרת, אם התא נמצא בקצה השמאלי של המערך, האיבר העוקב שלו נמצא בראש האלכסון הבא (בכיוון מטה). למשל, בדוגמא למעלה, העוקב של תא (1, 0) הוא (0, 2).
3. אחרת, העוקב של התא נמצא באלכסון ממנו – בכיוון למטה ושמאלה. למשל, בדוגמא למעלה, העוקב של תא (1, 1) הוא התא (2, 0).

שאלה 1 - 50 נקודות

המחלקה Cell הבאה מייצגת מספר שורה ועמודה של תא (איבר) **אחד** במערך הדו-ממדי.

עליכם לכתוב את המחלקה Cell. התחלנו אותה עבורכם, ואתם צריכים לממש את השיטות nextCell ו- prevCell המוגדרות להלן.

```
public class Cell{
     private int _currRow, _currCol;
     public Cell(int currRow, int currCol){
           _currRow = currRow;
           _currCol = currCol;
     }
     public void setCurrRow(int currRow){_currRow = currRow;}
     public void setCurrCol(int currCol) {_currCol = currCol;}
     public int getCurrRow () {return _currRow;}
     public int getCurrCol()  {return _currCol;}
     public  Cell nextCell(int rows, int columns) {
      // you have to implement
      }
     public  Cell prevCell(int rows, int columns) {
      // you have to implement
      }
}
```

התכונות _currRow ו _currCol מייצגות את השורה והעמודה של התא אותו מייצג האובייקט.

השיטה nextCell מחזירה את התא (cell) שנמצא **אחרי** התא עליו מופעלת השיטה, לפי הסדר האלכסוני המוגדר לעיל, כאשר הפרמטרים rows, columns הם מספר השורות והעמודות במטריצה (בהתאמה). אם נפעיל את השיטה nextCell על התא האחרון במערך הדו-ממדי (שורה אחרונה ועמודה אחרונה) יוחזר null.

השיטה prevCell מחזירה את התא (cell) שנמצא **לפני** התא עליו מופעלת השיטה, לפי הסדר האלכסוני המוגדר לעיל, כאשר הפרמטרים rows, columns הם מספר השורות והעמודות במטריצה (בהתאמה). אם נפעיל את השיטה prevCell על התא הראשון במערך הדו-ממדי (שורה 0 ועמודה 0) יוחזר null.

לדוגמא, בהינתן האובייקט הבא -

Cell c = new Cell(1, 2);

הקריאה לשיטה `c.nextCell(5, 5)` תחזיר את התא שבמקום (1, 2) והקריאה לשיטה `c.prevCell(5, 5)` תחזיר את התא שבמקום (3, 0).

שימו לב, השיטות nextCell, prevCell צריכות להחזיר תא **חדש** עם הערכים המתאימים.

כפתרון מיטבי לשאלה שיקבל את מלוא הנקודות יחשב פתרון פשוט ככל האפשר. לכן, נסו להבין את החוקיות והימנעו מתנאים מיותרים.
כמו כן,פתרון שישתמש בלולאות לא יקבל את מלוא הנקודות.

שאלה 2 - 50 נקודות

נתונה המחלקה Matrix, המייצגת מטריצה של מספרים שלמים.

```
public class Matrix {
    private int [][] _mat;

    public Matrix(int [][] mat) {
        _mat = new int [mat.length][mat[0].length];
        for (int i = 0; i<mat.length; i++)
            for (int j=0; j<mat[0].length; j++)
                _mat[i][j] = mat[i][j];
    }
}
```

א. (5%)

עליכם להוסיף למחלקה Matrix את השיטה toString המחזירה מחרוזת עם איברי המטריצה לפי שורות ועמודות.

כאשר בין מספר למספר באותה שורה מפריד Tab (על-ידי "\t") וכל שורה במטריצה מופיעה בשורה חדשה בפלט.

יש להקפיד על פורמט הפלט בדיוק כפי שמתואר – ללא טקסט או רווחים מיותרים!

ב. (15%)

עליכם להוסיף למחלקה Matrix שיטה הבודקת אם המטריצה ממוינת בסדר עולה (כלומר, מהקטן לגדול) לפי הדפוס האלכסוני שהוגדר לעיל (כלומר, להיעזר במחלקה Cell שכתבתם לעיל).

חתימת השיטה תהיה:

public boolean isSorted()

לדוגמא, המטריצה הימנית שלהלן ממוינת בסדר עולה בעוד שהמטריצה השמאלית לא ממוינת. (המספר 5 בשורה הראשונה במטריצה השמאלית הוא זה שמקלקל את המיון).

-18	-10	-2	**5**	7
-9	-1	2	7	14
0	6	7	16	19
6	12	18	29	37

-18	-10	-2	1	7
-9	-1	2	7	14
0	6	7	16	19
6	12	18	29	37

ג. (30%)

עליכם להוסיף למחלקה Matrix שיטה המסדרת את המטריצה כך שכל האיברים השליליים שבה יהיו בתחילת המטריצה לפי הדפוס האלכסוני שהוגדר לעיל, ולאחריהם המספרים החיוביים שבמטריצה. אפסים נחשבים לחיוביים בשאלה זו. אין צורך להפריד בינם ובין החיוביים.

הסדר בין המספרים החיוביים לבין עצמם ובין המספרים השליליים לבין עצמם אינו חשוב.

שימו לב, עליכם לדאוג שהשיטה תהיה **יעילה** – היא צריכה לעבור על איברי המטריצה פעם אחת בלבד! פתרון שיעבור יותר מפעם אחרת על איברי המטריצה יקבל ניקוד חלקי בלבד.

חתימת השיטה תהיה:

```
public void arrangeBySign()
```

לדוגמא, המטריצה הימנית תהפוך לאחר הסידור להיות המטריצה השמאלית

8	1	-2	-3	5
-9	7	4	-5	10
-8	15	2	6	-7
-6	12	-4	9	7

-7	-4	-2	-3	5
-9	-5	-6	7	10
-8	15	2	6	8
4	12	1	9	7

אסור להשתמש במערכי עזר!

ניתן להשתמש בשיטות עזר פרטיות ככל הנדרש.

שימו לב, בכל השיטות בשתי השאלות עליכם לבדוק את כל מקרי הקצה.

שימו לב, סטודנטים שלא הצליחו להגיע לפתרון של שאלה 1, יכולים להשתמש במחלקה `Cell` ששמנו באתר ולפתור בעזרתה את שאלה 2.

להזכירכם - יש לכתוב תיעוד פנימי ו- API

הגשה

1. הגשת הממ"ן נעשית בצורה אלקטרונית בלבד, דרך מערכת שליחת המטלות.
2. עליכם להגיש את הקבצים `Matrix.java`, `Cell.java`. עטפו אותם בקובץ zip ושלחו. אין לשלוח קבצים נוספים.