

## מבוא לחישוב- מטלה 3: רקורסיה, מערכים דו-מימדיים ומחרוזות

**שלב ראשון:** עליכם לממש את המחלקה Ex3 שתכיל את הפונקציות המתוארות להלן.

1

כתבו פונקציה **רקורסיבית** בשם `moveCharToLast` המקבלת מחרוזת `str` ותו `ch`, ומזיזה את כל המופעים של `ch` בתוך `str` לסוף `str`. לדוגמה, אם נתונה המחרוזת "hello" והתו 'l', המחרוזת שתוחזר תהיה "heoll".

חתימת הפונקציה:

```
public static String moveCharToLast(String str, char ch)
```

2

כתבו פונקציה **רקורסיבית** בשם `reduce` שמקבלת מחרוזת ומחזירה אותה "מצומצמת" – כל רצף של אותיות זהות הופך לאות אחת. לדוגמה, אם נתונה המחרוזת "aaabbcccccxyzz" המחרוזת שתוחזר תהיה "abcxyz".

חתימת הפונקציה:

```
public static String reduce(String str)
```

3

כתבו פונקציה בשם `mySplit` שמקבלת מערך של `int` המכיל מספרים חיוביים, וקוראת לפונקציית עזר **רקורסיבית**. הפונקציה `MySplit` צריכה לבדוק האם ניתן לחלק את המספרים במערך ל-2 קבוצות, כך ש-  
א. סכום המספרים בכל קבוצה יהיה זהה.  
ב. כל מספר יופיע באחת משתי הקבוצות.  
ג. כל המספרים שהם כפולה של 5 חייבים להיות באותה קבוצה.  
ד. כל המספרים שהם כפולות של 3 (ולא כפולות של 5) חייבים להיות בקבוצה השנייה.

ניתן להניח שמערך הקלט מכיל לפחות 2 איברים. לדוגמה, אם המערך מכיל את [5,21,8,15,7] הפונקציה תחזיר `true`, כי ניתן לחלק את המספרים ל-2 הקבוצות הבאות: {5,15,8} ו- {21,7}.

חתימת הפונקציה:

```
public static boolean mySplit(int[] nums)
```

4

השכנים של כל איבר במטריצה הם כל האיברים הסמוכים לאיבר זה: מעל, מתחת, בצדדים ובאלכסון. לכן, לכל איבר ייתכנו לכל היותר 8 שכנים, ולאיברים שבפינות יש רק 3 שכנים. כתבו פונקציה אשר תקבל כפרמטר מטריצה של `int`, ותחזיר מטריצה של `int` בה כל איבר יכיל את הסכום של כל השכנים שהיו לו במטריצה שניתנה כפרמטר. ניתן להניח שמטריצת הקלט היא לפחות 2X2. לדוגמה, אם נתונה המטריצה:

3	5	7	5
-4	2	10	11
9	-50	3	60

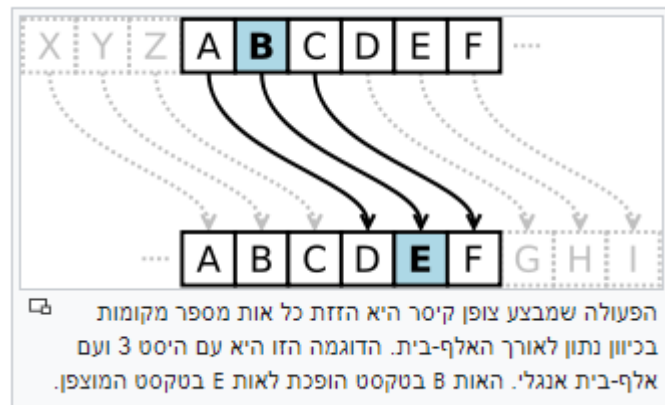
המטריצה שתוחזר תהיה:

3	18	33	28
-31	-17	43	85
-52	20	33	24

חתימת הפונקציה:

```
public static int[][] sumOfNeighbours(int[][] mat)
```

בתחום הקריפטוגרפיה, צופן קיסר, הידוע גם כצופן היסט, הוא אחד הצפנים הפשוטים בעולם ההצפנה. זהו סוג של צופן החלפה שבו כל אות בטקסט מוחלפת על ידי אות הנמצאת בהיסט קבוע כלשהו ממנה באלף-בית. ההיסט הקבוע הוא מפתח ההצפנה. למשל אם נקבע את ההיסט להיות 3, האות A תוחלף באות D, האות B תוחלף באות E, וכך הלאה באופן מחזורי (ז"א האות X תוחלף באות A). הכינוי קיסר נובע מכך שיוזיס קיסר נהג להשתמש בצופן על מנת לתקשר עם מפקדיו. כמו כל צופן החלפה אלף-ביתי, צופן קיסר מפוענח בקלות ולא מספק כמעט שום הגנה על אמצעי התקשורת.



כתבו פונקציה בשם caesarCipherText שמקבלת מחרוזת ומפתח הצפנה, ומחזירה את המחרוזת המוצפנת. ניתן להניח שהקלט המתקבל הוא של אותיות קטנות המהוות את ה-abc או תו רווח, אותו לא נצפין. במידה והמפתח אינו חוקי יש להחזיר מחרוזת ריקה. לדוגמה, אם נצפין את המחרוזת "defend the east wall of the castle" עם מפתח 1, נקבל "efefoe uif fbtu xbm m pg uif dbtumf". כדי לפענח את המחרוזת המוצפנת נוכל פשוט להשתמש בפונקציית ההצפנה עם מפתח שלילי.

חתימת הפונקציה:

```
public static String caesarCipherText(String str, int key)
```

צופן ויז'נר הוא צופן החלפה מתוחכם יותר, מכיוון שהוא מחליף כל אות במסר באות אחרת מתוך אלפבית שונה. ז"א המפתח הינו מילה קוד (או משפט) ומשתמשים בו באופן מחזורי. כל אות במפתח קובעת באיזה היסט יש להצפין כל אות במסר: האות a קובעת שיש להיסט 0 מקומות, האות b קובעת שיש להיסט 1 מקום אחד, האות c קובעת שיש להיסט 2 מקומות, וכן הלאה. לכן, בצופן ויז'נר יכול להיות שנצפין אותה של המסר בכל פעם באמצעות אות אחרת. לדוגמה, אם נצפין את המחרוזת "impressive student from ariel university" עם המפתח "honor", נקבל "pacfvzgvjv ggiulbg wycz rywz bbvjvygvhp". למה? את האות הראשונה במסר, i, נצפין באמצעות האות h במפתח, ז"א יש להיסט את i 7 מקומות קדימה. לכן, האות בצופן תהיה p. את האות השניה במסר, m, נצפין באמצעות האות o במפתח, ז"א יש להיסט את m 14 מקומות קדימה. לכן האות בצופן תהיה a, וכן הלאה. שימו לב שהאות s הראשונה במילה impressive הוצפנה באמצעות האות h של המפתח והאות s השניה הוצפנה באמצעות האות o של המפתח. לכן ה-s הראשונה הוצפנה לאות z וה-s השניה הוצפנה לאות g. בנוסף, שימו לב שלמרות שאנחנו לא מצפינים רווחים אנחנו כן מחשיבים אותם כאשר אנחנו חוזרים על המפתח, ולכן האות s במילה student הוצפנה באמצעות האות o של המפתח (ולא באמצעות האות h).

צופן ויז'נר מיוחס לדיפלומט הצרפתי בלז דה ויז'נר (1586), אולם למעשה הומצא מספר שנים לפניו על ידי הקריפטוגרף האיטלקי ג'ובני בטיסטה בלאסו (בשנת 1553). למרות

שהצופן לא משמר את תדירות האותיות במסר המקורי הוא עדיין נחשב לא בטוח לשימוש מכיוון שהוא חשוף לניתוח סטטיסטי.

עליכם לכתוב בשאלה זו 2 פונקציות. הראשונה מצפינה באמצעות צופן ויזנר, ולכן היא מקבלת מחרוזת ומפתח הצפנה ומחזירה את המחרוזת המוצפנת. השניה מפענחת את צופן ויזנר, ולכן היא מקבלת מחרוזת מוצפנת ומפתח הצפנה ומחזירה את המסר המפוענח. גם בשאלה זו ניתן להניח שהקלט המתקבל הוא של אותיות קטנות המהוות את ה- abc או תו רווח, אותו לא נצפין. ניתן להניח שהמפתח הוא מחרוזת לא ריקה.

חתימת הפונקציות:

```
public static String vigenereCipherText(String str, String key)
public static String vigenereDecipherText(String str, String key)
```

**שלב שני:** כתבו מחלקת בדיקה בעזרת Junit 5 בשם Ex3Test. תוכלו לעשות שימוש במחלקה המצורפת אך עליכם להרחיב אותה כך שהיא תאפשר בדיקה של כל המקרים הרלוונטיים.

**שלב שלישי:** ייצרו קובץ דחוס בשם Ex3.zip שמכיל אך ורק את הקבצים Ex3.java ו Ex3Test.java, והעלו אותו לתיבת ההגשה במודל. הקפידו לרשום את מספר תעודת הזהות שלכם בתיעוד של כל אחת מהמחלקות. בנוסף, כדי להקל על הבדיקה, עליכם לכתוב פונקציה `public static String myID()`, שמחזירה את מספר תעודת הזהות שלכם.

הנחיות כלליות למטלות:

1. את המטלות יש לעשות לבד, בכל מקרה של שימוש בקוד או במידע מגורמים אחרים יש לציין זאת באופן מפורש במטלה עצמה.
2. חובה לתעד את המטלה, כל רכיב בתוכנית (פונקציה, מחלקה וכו') מחויב בתיעוד מסודר.
3. את המטלה יש להגיש במועד למערכת הגשת המטלות במודל.