תוכנה 1 – חורף תשע"ו

תרגיל מספר 3

מערכים ומחרוזות

<u>הנחיות כלליות:</u>
קראו בעיון את קובץ נהלי הגשת התרגילים אשר נמצא באתר הקורס.

- הגשת התרגיל תיעשה במערכת ה-moodle בלבד (http://moodle.tau.ac.il/).
- יש להגיש קובץ zip יחיד הנושא את שם המשתמש ומספר התרגיל (לדוגמא, עבור המשתמש aviv יקרא הקובץ aviv_hw3.zip). קובץ ה-zip יכיל:
  א. קובץ פרטים אישיים בשם details.txt המכיל את שמכם ומספר ת.ז.
  ב. קבצי ה- java של התוכניות אותם התבקשתם לממש.

הערות כלליות:

- הקפידו שחתימות המתודות תהיינה **<u>זהות</u>** לאלו המצוינות בשאלה.
- ניתן להוסיף מתודות עזר.
- בתרגיל זה אין צורך לטפל במקרים בהם מערכי\מחרוזות הקלט ריקים או שווים ל-null אלא אם צוין אחרת.

חלק א' – מערכים

ממשו מחלקה בשם **ArrayUtils** שתכיל את המתודות הסטטיות הבאות:

<u>בחלק זה מבנה הנתונים היחידי בו מותר להשתמש הינו מערכים.</u>

1. **[10 נק']** ממשו מתודה בשם ***shiftArrayToTheRight*** המקבלת מערך המכיל מספרים שלמים ומחזירה מערך חדש בו איברי מערך הקלט מוזזים לימין כמספר הפעמים שצוין. האיבר האחרון במערך בקלט מוזז אל האיבר הראשון במערך הפלט. ניתן להניח כי המערך המתקבל אינו null, אך יש לוודא כי מספר ההזזות חוקי (חיובי). במידה ומספר ההזזות אינו חוקי (שלילי),המתודה תחזיר את המערך ללא שינוי.

חתימת המתודה:

```
public static int[] shiftArrayToTheRight(int[] array, int move)
```

דוגמא:

```
shiftArrayToTheRight([1, 2, 3, 4, 5],1) -> [5, 1, 2, 3, 4]

shiftArrayToTheRight([1, 2, 3, 4, 5],3) -> [3, 4, 5, 1, 2]
```

2. **[10 נק']** ממשו מתודה בשם ***matrixTrace*** המקבלת מערך דו מימדי, ומחזירה את סכום האיברים אשר נמצאים על האלכסון הראשי. ניתן להניח כי המערך המתקבל הוא מטריצה ריבועית, אשר איננה null.

חתימת המתודה:

public static int matrixTrace (**int**[][] m)

דוגמא:

matrixTrace ([[1, 2, 3], [1, 2 , 3], [1, 2, 3]]) -> 6 //(1+2+3)

matrixTrace ([[1,3,4], [1, 2,5], [1, 2, 3]]) -> 6 //(1+2+3)

3. א. **[10 נק']** ממשו מתודה בשם ***matrixSwitchRows*** המקבלת מערך דו מימדי המכיל מספרים שלמים, המייצג מטריצה מלבנית (אורך כל שורה הוא זהה), ושני אינדקסים ומבצעת החלפה בין שתי השורות. ניתן להניח כי האינדקסים המתקבלים אכן חוקיים. במידה ומתקבל אותו האינדקס, המתודה לא תעשה דבר. המתודה מחזירה את המטריצה לאחר ההחלפה. ניתן להניח כי המערך אינו null.
חתימת המתודה:

public static int [][] matrixSwitchRows (**int**[][] m, int I, int j)

דוגמא:

matrixSwitchRows ([[1,2,3],[4,5,6],[7,8,9]],0,2) -> [[7,8,9],[4,5,6],[1,2,3]]

matrixSwitchRows ([[1,2,3],[4,5,6],[7,8,9]],1,1) -> [[1,2,3],[4,5,6],[7,8,9]]

matrixSwitchRows ([[1,2],[4,5],[7,8]],2,1) -> [[1,2],[7,8],[4,5]]

ב. **[10 נק']** ממשו מתודה בשם ***matrixScalarRow*** המקבלת מערך דו מימדי המכיל מספרים שלמים, המייצג מטריצה מלבנית (אורך כל שורה הוא זהה), ושני int . ה-int הראשון מייצג סקאלר, והשני את אינדקס השורה אותה יש להכפיל בסקלאר. ניתן להניח כי האינדקס המתקבל אכן חוקי. המתודה מחזירה את המצטריצה לאחר ההכפלה. ניתן להניח כי המטריצה איננה null.
חתימת המתודה:

public static int [][] matrixScalarRow (**int**[][] m, int s, int j)

דוגמא:

matrixScalarRow ([[1,2,3],[4,5,6],[7,8,9]],0,2) -> [[7,8,9],[4,5,6],[0,0,0]]

matrixScalarRow ([[1,2,3],[4,5,6],[7,8,9]],2,1) -> [[1,2,3],[8,10,12],[7,8,9]]

matrixScalarRow ([[1,2],[4,5],[7,8]],2,1) -> [[1,2],[8,10],[7,8]]

4. **[20 נק']** ממשו מתודה בשם ***matrixMultiplication*** המקבלת שני מערכים דו מימדיים המכיל מספרים שלמים, ומייצגים מטריצות מלבניות, ומחשבת את כפל המטריצות. לקריאה כיצד לחשב זאת קראו: כפל מטריצות.
המתודה מחזירה את התוצאה. ניתן להניח כי אורכי המטריצות עומדים בדרישות לביצוע הכפל, וכי המטריצות שונות מ-null.

חתימת המתודה:

public static int [][] matrixMultiplication (**int**[][] m, int[][] n)

דוגמא:

matrixMultiplication ([[1,2,3],[4,5,6],[7,8,9]],[[1,0,0],[0,1,0],[0,0,1]]) ->
[[1,2,3],[4,5,6],[7,8,9]]

matrixMultiplication ([[1,2],[3,4],[5,6]] ,[[1,1],[2,2]]) -> [[5,5],[11,11],[17,17]]

חלק ב' – מחרוזות

ממשו מחלקה בשם **StringUtils** שתכיל את המתודות הסטטיות הבאות:

5. **[10 נק']** ממשו מתודה בשם **sortStringWords** המקבלת מחרוזת קלט (הכוללת אותיות אנגליות ורווחים בלבד), ומחזירה מחרוזת בה מופיעות המילים ממחרוזת הקלט כשהן ממוינות לקסיקוגרפית בסדר יורד (עם רווחים בין המילים). ניתן להניח כי המחרוזת המתקבלת שונה מ-null.
   - ✓ רמז: היעזרו בפקודה split של המחלקה String

חתימת המתודה:

public static String sortStringWords (String str)

דוגמא:

sortStringWords(“To Be Or Not To Be”) -> "To To Or Not Be Be"

6. **[15 נק']** ממשו מתודה בשם ***deleteSubString*** המקבלת שתי מחרוזות . במידה והמחרוזת הראשונה הינה תת מחרוזת של השניה, יש להחזיר מחרוזת אשר מתקבלת ממחיקת תת המחרוזת הראשונה מן השנייה. אחרת, יש להחזיר את המחרוזת השנייה ללא שינוי. ניתן להניח כי הקלט תקין, כלומר שהמחרוזות אינן null. ניתן להגדיר מתודות עזר לצורך המימוש . תת המחרוזת הראשונה הינה תת מחרוזת של השניה אם היא מופיעה כולה במחרוזת השנייה **ברצף.**
   - ✓ רמז: השתמשו במתודה: contains של המחלקה String

חתימת המתודה:

public static String deleteSubString(String sub, String s)

דוגמאות :

deleteSubString ("If you don’t have dreams, ","**If you don’t have dreams,** you’ll never make your dreams come true ") -> "you’ll never make your dreams come true"

deleteSubString ("It is better to be roughly wrong ","It is better to be roughly right than precisely wrong ") -> "It is better to be roughly right than precisely wrong"

deleteSubString ("Conversation"," Experience is simply the name we give our mistakes") -> " Experience is simply the name we give our mistakes"

deleteSubString ("s. I am always ","I have the simplest taste**s. I am always** satisfied with the best") -> "I have the simplest tastesatisfied with the best"

7. **[15 נק']** כתוב מתודה בשם: ***mergeStrings*** המקבלת שתי מחרוזות המורכבות מאותיות בלבד ללא הופעות חוזרות (אות יכולה להופיע בכל מחרוזת פעם אחת בלבד, אבל יכולה להופיע בשתיהן). המתודה תחזיר מחרוזת חדשה, המכילה רק את האותיות המופיעות גם במחרוזת

הראשונה וגם במחרוזת השניה. אין צורך לבדוק תקינות הקלט. במידה ואין תווים אשר מופיעים בשתי המחרוזות, יש להחזיר את המחרוזת הריקה. סדר התווים במחרוזת המוחזרת יקבע על ידי המחרוזת הראשונה.

חתימת המתודה:

```
public static String mergeStrings(String a, String b)
```

דוגמאות:

mergeStrings ("boy","girl") -> ""

mergeStrings ("catdog","boygirl") -> "og"

mergeStrings ("abcdefg","bcgfhi") -> "bcfg"

בהצלחה !