

תוכנה 1 – אביב תשע"ה

תרגיל מספר 3

מערכים ומחרוזות

הנחיות כלליות:

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 יכול:
 - א. קובץ פרטים אישיים בשם details.txt המכיל את שמכם ומספר ת.ז.
 - ב. קבצי ה-java של התוכניות אותם התבקשתם לממש.
 - ג. קובץ PDF בשם answers.pdf המכיל את התשובות לשאלות.

הערות כלליות:

- הקפידו שחתימות המתודות תהיינה זהות 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.
- היעזרו בתוכנית ההדגמה ובפלט לדוגמא המצורפים לתרגיל.

חלק א' – מערכים

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

1. ממשו מתודה בשם **reverseArray** המבצעת היפוך של מערך מספרים נתון. המתודה מחזירה מערך חדש המכיל את האיברים בסדר הפוך.

חתימת המתודה:

```
public static int[] reverseArray(int[] array)
```

דוגמא:

```
reverseArray([1, 2, 3, 4, 5, 6]) -> [6, 5, 4, 3, 2, 1]
```

2. ממשו מתודה בשם **fillStringArray** המקבלת מערך מחרוזות, ומערך של מספרים שלמים. המתודה תחזיר מערך חדש של מחרוזות המורכב מהמחרוזות שבמערך הקלט הראשון, כאשר כל מחרוזת משוכפלת לפי המספר המופיע בתא המקביל במערך השני.

חתימת המתודה:

```
public static String[] fillStringArray (String[] strs, int[] repeats)
```

דוגמאות:

```
fillStringArray ([aa, bb, cc, dd], [1, 2, 3, 4]) -> [aa, bb, bb, cc, cc, cc, dd, dd, dd, dd]
```

3. ממשו מתודה בשם **transposeMatrix** המחזירה מטריצה משוחלפת בה האיבר ה-(i,j) שווה בערכו לאיבר ה-(j,i) במטריצת הקלט. ניתן להניח שמטריצה הקלט היא מלבנית (כל השורות בה הן באותו אורך).

חתימת המתודה:

```
public static int[][] transposeMatrix(int[][] matrix)
```

דוגמאות:

```
m = [[1, 2, 3, 4],
      [5, 6, 7, 8],
      [9, 10, 11, 12]];
```

```
transposeMatrix(m) -> [[1, 5, 9],
                       [2, 6, 10],
                       [3, 7, 11],
                       [4, 8, 12]]
```

```
n = [[1, 1, 1],
      [2, 2, 2],
      [3, 3, 3]];
```

```
transposeMatrix(n) -> [[1, 2, 3],
                       [1, 2, 3],
                       [1, 2, 3]]
```

4. ממשו מתודה בשם **splitArrayByNum** המקבלת מערך מספרים שלמים, ומספר שלם שימש כמפריד (Delimiter), ומפצלת את מערך הקלט למערכים המופרדים ע"י המספר המפריד.
המתודה תחזיר מערך דו מימדי של מספרים שלמים, כשכל שורה היא רצף מספרים ממערך הקלט המקורי שאינו כולל את המספר המפריד.
✓ רמז: השתמשו במתודה `Arrays.copyOf`.
✓ שימו לב למקרי הקצה בהם המספר המפריד מופיע בתחילת המערך, בסוף המערך, או ברצף.

חתימת המתודה:

```
public static int[][] splitArrayByNum(int[] input, int number)
```

דוגמאות:

```
splitArrayByNum([0, 0, 0, 3, 1, 1, 1], 3) -> [[0, 0, 0], [1, 1, 1]]
```

```
splitArrayByNum([1, 2, 3, 1, 2, 3, 1, 1, 2, 2, 3, 1], 3) -> [[1, 2], [1, 2], [1, 1, 2, 2], [1]]
```

```
splitArrayByNum([3, 1, 3, 3], 3) -> [[1]]
```

5. ממשו מתודה בשם **matrixMean** המקבלת מערך דו מימדי, ומחזירה את הממוצע על כל איברי המערך. שימו לב, בשאלה זו יתכן וכל שורה במערך היא באורך שונה. ניתן להניח שמערך הקלט אינו ריק.

חתימת המתודה:

```
public static double matrixMean (int[][] m)
```

דוגמא:

```
matrixMean ([[1, 2, 3], [1, 2, 3], [1, 2, 3]]) -> 2.0
```

```
matrixMean ([ [1], [1, 2], [1, 2, 3]]) -> 1.666667
```

חלק ב' – מחרוזות

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

6. ממשו מתודה בשם **isPalindrome** המקבלת מחרוזת תווים ומחזירה true אם המחרוזת מייצגת רצף תווים פלינדרומי ו- false אחרת. **פלינדרום** יוגדר כמילה שקריאתה מימין לשמאל או משמאל לימין תניב את אותו רצף תווים.

```
public static boolean isPalindrome (String str)
```

דוגמא:

```
isPalindrome("abccba") -> true
```

```
isPalindrome("abcdcba") -> true
```

```
isPalindrome("abcdba") -> false
```

7. ממשו מתודה בשם **sortStringWords** המקבלת מחרוזת קלט (הכוללת אותיות אנגליות ורווחים בלבד), ומחזירה מחרוזת בה מופיעות המילים ממחרוזת הקלט כשהן ממוינות לקסיקוגרפית בסדר עולה (עם רווחים בין המילים). המיון צריך להיות case-sensitive. ✓
רמז: היעזרו בפקודה split של המחלקה String, וב- Arrays.sort.

חתימת המתודה:

```
public static String sortStringWords (String str)
```

דוגמא:

```
sortStringWords("To Be Or Not To Be") -> "Be Be Not Or To To"
```

8. ממשו מתודה בשם **isStringArraySorted** המקבלת מערך של מחרוזות ומספר שלם. המתודה תחזיר ערך בוליאני המציין האם המחרוזות במערך הקלט ממוינות לקסיקוגרפית, כאשר היא מתחשבת רק ב-ח התווים הראשונים בכל מחרוזת.

✓ ניתן להניח שבכל מחרוזת מאוחסנת מילה אחת בלבד באנגלית, ללא רווחים, ושאוּרְכָה גדול מ-ח.

✓ יש לבצע את החישובים במצב Case-insensitive.

חתימת המתודה:

```
public static boolean isArraySorted(String[] strs, int n)
```

דוגמאות:

```
isArraySorted ([Alpha, bravo, Charlie, Delta, Echo],4) -> true  
isArraySorted ([Bravo, Alpha, Charlie, Delta, Echo],4) ->false  
isArraySorted ([aaaC, aaaB, BBBb, BBBa, CCC],3) -> true
```

9. ממשו את המתודה **stringHistogram** הבונה היסטוגרמה של התווים במחרוזת. המתודה מקבלת מחרוזת המכילה את האותיות a עד z בלבד, ומחזירה מערך של שלמים באורך 26 כך שערך התא ה-i הוא מספר המופעים של התו 'a'+i במחרוזת.

חתימת המתודה:

```
public static int[] stringHistogram(String a)
```

דוגמאות:

```
stringHistogram("bannana") -> [3, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]  
stringHistogram("missisipi") -> [0, 0, 0, 0, 0, 0, 0, 0, 4, 0, 0, 0, 1, 0, 0, 1, 0, 0, 3, 0, 0, 0, 0, 0, 0]
```

10. ממשו מתודה בשם **areAnagrams** המקבלת שתי מחרוזות ובודקת האם האחת מתקבלת משיכול אותיות השנייה. ניתן להניח שהקלט מורכב רק מאותיות באנגלית ומרווחים, כאשר לצורך שאלה זו, גודל האותיות והרווחים אינם משנים (כלומר "A b" נחשב לשיכול אותיות של "Ba"). ניתן להניח כי הקלט תקין, כלומר שהמחרוזות אינן null. ניתן להגדיר מתודות עזר לצורך המימוש.

חתימת המתודה:

```
public static boolean areAnagrams(String a, String b) -> false
```

דוגמאות :

```
areAnagrams("Debit Card","Bad Credit") ->true  
areAnagrams("A decimal point","Im a dot in place") -> true  
areAnagrams("Conversation","Voices rant on") ->true  
areAnagrams("Radar","Tartar") ->false
```

בהצלחה !