

תוכנה 1 – אביב תשע"ד

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 יכיל:
 - 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 מערכים/מחרוזות הקלט ריקים או שווים ל-null אלא אם צוין אחרת.

חלק א' – מערכים

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

1. ממשו מתודה בשם **shiftArrayToTheRight** המקבלת מערך המכיל מספרים שלמים ומחזירה מערך חדש בו איברי מערך הקלט מוזזים מקום אחד לימין. האיבר האחרון במערך בקלט מוזז אל האיבר הראשון במערך הפלט.

חתימת המתודה:

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

דוגמא:

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

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)
```

דוגמא:

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

3. ממשו מתודה בשם **mergeArrays** המקבלת כקלט שני מערכי מספרים, ומחזירה מערך חדש המכיל את איברי שני המערכים לסירוגין. במידה ואורך המערכים שונה, מערך הפלט יכיל את האיברים משני המערכים לסירוגין בהתחלה, ולאחר מכן יכיל ברצף את האיברים של המערך הגדול יותר.

חתימת המתודה:

```
public static int[] mergeArrays (int[] a, int[] b)
```

דוגמא:

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

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

4. 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
```

5. ממשו מתודה בשם **matrixMinColumnId** המקבלת מערך דו מימדי המכיל מספרים שלמים, מחשבת מינימום על המספרים בכל עמודה בנפרד, ומחזירה את אינדקס העמודה בעלת ערך המינימום המינימלי מבין כל העמודות במטריצה (העמודה אשר בה מצוי המספר הקטן ביותר במטריצה כולה). בשאלה זו ניתן להניח שהמטריצה היא ריבועית ושאינה ריקה.

חתימת המתודה:

```
public static int matrixMinColumnId (int[][] m)
```

לצורך מימוש המתודה, הגדירו מתודת עזר המקבלת מטריצה ואינדקס של עמודה במטריצה, ומחזירה את ערך המינימום של המספרים בעמודה זו.

חתימת המתודה:

```
private static int matrixColumnMin (int[][] m, int j)
```

דוגמא:

```
matrixColumnMin ([[1,2,3],[4,5,6],[7,8,9]],0) -> 1 // min of column 0 is 1
```

```
matrixColumnMin ([[1,2,3],[4,5,6],[7,8,9]],1) -> 2 // min of column 1 is 2
```

```
matrixCounmlMin ([[1,2,3],[4,5,6],[7,8,9]],2) -> 3 // min of column 2 is 3
```

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

```
// the index of the matrix column with minimal minimum value is 0
```

6. א. ממשו מתודה בשם **concatMatrixRows** המקבלת מערך דו מימדי המכיל מספרים שלמים, ומחזירה מערך חד מימדי המכיל את כל איברי מערך הקלט, כאשר השורות משורשות זו אחר זו.

בשאלה זו יש להניח שהמטריצה היא ריבועית ושאינה ריקה (כל השורות הן באותו אורך).

חתימת המתודה:

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

דוגמא:

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

ב. ממשו מתודה בשם **concatNonRectangleMatrixRows** המבצעת את אותה פעולה כמו המתודה בסעיף א', אך הפעם יתכן וכל שורה במטריצה היא באורך שונה, ריקה, או כלל אינה קיימת (null). בשאלה זו יש לייצר את מערך הפלט גם אם קיימות שורות ריקות או null במטריצת הקלט. ניתן להניח שמטריצת הקלט אינה ריקה לגמרי.

חתימת המתודה:

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

דוגמא:

```
concatMatrixRows ([[1,2,3],[4,5],[6],[],[null],[7,8]]) -> [1, 2, 3, 4, 5, 6, 7, 8]
```

חלק ב' – מחרוזות

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

7. ממשו מתודה בשם **reverseString** המקבלת מחרוזת תווים ומחזירה את היפוכה.

חתימת המתודה:

```
public static String reverseString (String source)
```

דוגמא:

```
reverseString ("Hello") -> "olleH"
```

8. ממשו מתודה בשם **sortStringWords** המקבלת מחרוזת קלט (הכוללת אותיות אנגליות ורווחים בלבד), ומחזירה מחרוזת בה מופיעות המילים ממחרזת הקלט כשהן ממוינות לקסיקוגרפית בסדר עולה (עם רווחים בין המילים).
✓ רמז: היעזרו בפקודה `split` של המחלקה `String`

חתימת המתודה:

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

דוגמא:

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

9. 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
```

10. 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, 4, 0, 0, 0, 1, 0, 0, 1, 0, 0, 3, 0, 0, 0, 0, 0, 0]
```

בהצלחה !