

# תוכנה 1 – אביב תשע"ג

## תרגיל מספר 3

### 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 טקסט answers.txt עם התשובות לשאלות (אפשר קובץ טקסט או קובץ Word).

שימו לב,

- לכל אורך התרגיל ניתן להוסיף שירותי עזר. הקפידו ששירותים אלה לא יפגעו בנדרש בסעיפים אחרים.
- כדי להקל עליכם, אנו מספקים מחלקות הבודקות את הדוגמאות אשר מפורטות בגוף התרגיל. אם הקוד שמימשתם עונה נכונה על הדוגמאות הנתונות, תתקבל הודעה "QX pass" כאשר X הוא מספר השאלה. אחרת, תתקבלנה הודעות שגיאה מתאימות.
- **חשוב:** הודעת הצלחה אינה מעידה על נכונות הקוד אלא על התאמה לדוגמאות הספציפיות אותן סיפקנו. לעומת זאת, הודעות שגיאה מעידות על כך שהמימוש שגוי.
- **חשוב מאוד:** קוד שאינו מתקמפל עם תוכניות הבדיקה לא ייבדק. בהצלחה

## 1. מערכים

יש לממש את המחלקה Assignment3Q1 אשר תכיל את השירותים הבאים:

- א. (10 נקודות) ממשו את השרות `reverseArray()` אשר הופך מערך של שלמים. השירות מחזיר מערך חדש כך שבתא ה-i במערך הפלט ימצא תוכנו של התא ה-1 - i - length ממערך הקלט. חתימת השירות:

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

דוגמאות:

```
reverseArray([1, 2, 3, 4, 5, 6]) -> [6, 5, 4, 3, 2, 1]  
reverseArray([4, 6, 2, 1, 3, 9, 9, 9]) -> [9, 9, 9, 3, 1, 2, 6, 4]
```

- ב. (10 נקודות) נגדיר פעולת סיבוב על מערך של שלמים בצורה הבאה: כל תא במערך הפלט מכיל את ערך התא העוקב לו במערך הקלט, למעט התא האחרון המכיל את ערך התא הראשון בקלט. ממשו את השירות `rotateArray` המחזיר מערך חדש המכיל את סיבוב המערך המקורי. חתימת השירות:

```
public static int[] rotateArray(int[] arr)
```

דוגמאות:

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

ג. (10 נקודות) נגדיר פעולת פיזור המקבלת מערך ואינדקס  $i$ . הפעולה מפזרת את הערך הרשום בתא ה- $i$  (נסמנו ב- $k$ ) לתאים העוקבים ע"י איפוס התא ה- $i$  הגדלתם של התאים העוקבים ב-1. התא העוקב לתא האחרון במערך הוא התא הראשון. אינטואיטיבית: אם נחשוב על ערך התא כמייצג כמות של כדורים, הפעולה מפזרת את הכדורים ע"י לקיחתם מהתא המבוקש ופיזור כדור בכל תא עוקב. במקרה שהגענו לסוף המערך ונותרו לנו כדורים ממשיכים את התהליך מתחילת המערך. שימו לב: בשונה מהפעולות הקודמות שהוגדרו, פעולה זו משנה את תוכן מערך הקלט ואינה מחזירה דבר. חתימת השירות:

```
public static void spreadInArray(int[] data, int index)
```

דוגמאות:

```
int[] source = { 1, 2, 3, 4, 5, 6 };
spreadInArray(source, 2); // source == [1, 2, 0, 5, 6, 7]
                          // quantity of 3 from the 3rd cell was
                          // distributed to the 4th, 5th and 6th cells.

spreadInArray(source, 4): // source == [2, 3, 1, 6, 1, 8]
                          // quantity of 6 from the 5th cell was spread
                          // to the 6th cell, 1st cell (as the array size
                          // is six), 2nd, 3rd, 4th and 5th cell. Note that
                          // the 5th cell holds a quantity of 1 rather than
                          // zero).

spreadInArray(source, 0): // source == [0, 4, 2, 6, 1, 8]
                          // similar to the first case.
```

## 2.2. מחרזות

יש לממש את המחלקה Assignment3Q2 אשר תכיל את השירותים הבאים:

א. (10 נקודות) ממשו את השירות reverseString אשר הופך את סדר התווים במחרזת. חתימת השירות:

```
public static String reverseString(String a)
```

דוגמאות:

```
reverseString("Hello") -> "olleH"
reverseString("smartphone") -> "enohptrams"
reverseString("peep") -> "peep"
```

ב. (10 נקודות) ממשו את הפעולה `rotateString` אשר מחזירה מחרוזת בה כל התווים מוסטים ממקומם תו אחד לשמאל למעט התו הראשון העובר לסוף המחרוזת. חתימת השירות:

```
public static String rotateString(String a)
```

דוגמאות:

```
rotateString("abcdef") -> "bcdefa"
rotateString("software") -> "oftwares"
rotateString("oftwares") -> "ftwareso"
```

ג. (10 נקודות) ממשו את השירות `stringHistogram` הבונה היסטוגרמה של התווים במחרוזת. השירות מקבל מחרוזת המכילה את האותיות `a` עד `z` בלבד, ומחזיר מערך של שלמים באורך 26 כך שערך התא `i`-הוא מספר המופעים של התו `'a'+i` במחרוזת. חתימת השירות:

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

דוגמאות:

```
stringHistogram("bannana") -> [3, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0,
                                0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
stringHistogram("missisipi") -> [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 4, 0, 0, 0, 0, 1, 0,
                                0, 1, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
stringHistogram("gooooooooooogle") -> [0, 0, 0, 0, 1, 0, 2, 0, 0, 0, 0, 0, 0, 1,
                                       0, 0, 11, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

### 3. מטריצות

בשאלה זו נעסוק במערכים דו מימדיים של מספרים שלמים המתארים מטריצה מרובעת. כל איבר במערך הראשי, הוא מערך חד-מימדי המתאר שורה במטריצה. כל המתודות במחלקה זו ימומשו במחלקה `Assignment3Q3`.

א. (10 נקודות) ממשו את השירות `multiplyMatrix` המכפיל מטריצה בסקלר. על השירות לשנות את מטריצת הקלט. חתימת השירות:

```
public static void multiplyMatrix(int[][] matrix, int scalar)
```

דוגמאות:

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

multiplyMatrix(m, 3) // m = { {3, 6, 9, 12},
                              // {15, 18, 21, 24},
                              // {27, 30, 33, 36} }
```

ב. (10 נקודות) ממשו את השירות transposeMatrix המחזיר מטריצה משוחלפת בה האיבר ה-(i,j) שווה בערכו לאיבר ה-(j,i) במטריצת הקלט. חתימת השירות:

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

דוגמאות:

```
int[][] 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} }
```

```
int[][] n = { {1, 1, 1},
              {2, 2, 2},
              {3, 3, 3} };

transposeMatrix(n) -> { {1, 2, 3},
                       {1, 2, 3},
                       {1, 2, 3} }
```

ג. (20 נקודות) ממשו את השירות hasThree המחזיר true אם ישנם שלושה ערכים שווים במטריצה המופיעים בתאים עוקבים לאורך שורה, עמודה או אלכסון ו-false אחרת. הדרכה: ממשו את בדיקת המקרים השונים (שורה, עמודה אלכסון) בשירותים נפרדים והפעילו אותן מהשירות אותו נתבקשתם לממש. שימו לב שבחלק מן המקרים יש להגביל את הלולאות ולא לעבור על כל המטריצה. חתימת השירות:

```
public static boolean hasThree(int[][] matrix)
```

דוגמאות (לנוחיותכם, בכל דוגמא המחזירה true הודגשו הערכים המתאימים).

```
int[][] m = { {1, 1, 1, 2},
              {3, 4, 5, 6},
              {7, 8, 9, 0} };

hasThree(m) -> true
```

```
int[][] m = { {1, 2, 1, 2},
              {1, 4, 5, 6},
              {1, 8, 9, 0} };

hasThree(m) -> true
```

```
int[][] m = { {1, 2, 1, 2},
              {4, 1, 5, 6},
              {8, 9, 1, 0} };

hasThree(m) -> true
```

```
int[][] m = { {1, 2, 1, 2},  
              {4, 1, 5, 6},  
              {1, 8, 9, 0} };  
hasThree(m) -> true
```

```
int[][] m = { {1, 2, 5, 2},  
              {4, 1, 5, 6},  
              {1, 8, 9, 0} };  
hasThree(m) -> false
```

## בהצלחה