

# תוכנה 1 – סתיו תשע"ד

## תרגיל מספר 4

### הנחיות כלליות:

- קראו בעיון את קובץ נהלי הגשת התרגילים אשר נמצא באתר הקורס.
- הגשת התרגיל תעשה 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\_hw4.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.

בתרגיל זה ניצור מחלקות העוסקות במשחקי מילים. בפרט, נעבוד עם מערכים, מחרוזות, והמחלקה Scanner. כאשר אתם כותבים קוד, חישבו תחילה כיצד ניתן לפרקו למשימות קטנות ופשוטות יותר, וכתבו מתודות ומחלקות עזר בהתאם. עליכם לכתוב את כל המחלקות בתוך החבילה `il.ac.tau.cs.sw1.wordplay`

**הערה כללית:** בתרגיל זה מצופה מכם לבדוק את חוקיות הקלט לתוכניות שאתם כותבים. למשל, מס' הארגומנטים שהתקבלו משורת הפקודה. אם הקלט לא חוקי, יש להדפיס הודעה בהתאם. עם זאת:

- אם מצוין בסעיף כלשהו כי ניתן להניח משהו על הקלט, אין צורך לבדוק זאת בקוד של אותו סעיף (הנחת קדם). לדוגמא, בסעיף (I)2 אין צורך לבדוק את חוקיות מחרוזת התבנית שמועברת כקלט, אבל בסעיף (II)2 יש לבדוק זאת.
- בפונקציות בהן מתקבל כקלט מסלול לקובץ, אין צורך לבדוק את חוקיות המסלול או אם הוא מצביע לקובץ קיים. כמו כן, אין צורך לטפל בשגיאות הקשורות בקבצים בשלב זה, אלא להשתמש ב-`throws` כפי שראיתם בכיתה.

### 1. [25 נק']

(I) כתבו את מחלקת העזר `WordPool` אשר תשמש ליצירת מאגר מילים מקובץ טקסט, לצורכי המשחקים שלנו. בתוך המחלקה הגדירו את השירות הבא:

```
public static String[] createPoolFromScanner(Scanner scanner)
```

השירות יקבל כקלט עצם מסוג `java.util.Scanner` ויניח שהוא כבר מאותחל לקריאה ממקור כלשהו (למשל, קובץ). עליכם לקרוא את המילים בעזרת ה-`Scanner` ולהחזיר מערך עם כל המילים שנקראו **ללא חזרות**.

- עליכם להמיר תחילה את כל המילים שתקראו ל-`lowercase`.
- אם מילה שקראתם היא באורך 0, או מכילה תו שאינו אות בא"ב האנגלי (a-z), היא אינה "חוקית" ועליכם להתעלם ממנה.
- יש להחזיר רק את 2000 המילים ה"חוקיות" השונות הראשונות. בפרט, גודל המערך המוחזר לא יעלה על 2000.

- אם בסיום הקריאה מצאתם פחות מ- 2000 מילים שונות, גודל המערך המוחזר צריך להיות בהתאם. למשל, אם קראתם 1177 מילים, החזירו מערך בגודל 1177 שמכיל אותן.

(II) הוסיפו פונקציה main במחלקה WordPool, אשר מקבלת מסלול לקובץ כארגומנט בשורת הפקודה, יוצרת Scanner שקורא מן הקובץ הזה, קוראת ל- createPoolFromScanner שכתבתם כדי לחשב את מאגר המילים ואז מדפיסה כל מילה בשורה נפרדת. ניתן לבדוק את עצמכם ע"י הרצת התכנית על הקובץ story.txt המצורף לתרגיל. בקובץ זה יש 1177 מילים חוקיות שונות.

## 2. [35 נק']

(I) כעת, ניצור מחלקה בשם HangmanHelper, אשר מטרתה לסייע במשחק "האיש התלוי", בו יש לנחש מילה כאשר חלק מאותיותיה אינן ידועות. עליכם לממש בתוך המחלקה את השירות הבא:

```
public static String[] suggestCompletion(String[] pool, String pattern)
```

שירות זה יקבל כקלט רשימת מילים ומחרוזת "תבנית". תבנית מורכבת מאותיות a-z וכן ממקפים תחתיים (underscores). השירות יחזיר מערך המכיל את כל המילים ברשימה אשר מתאימות לתבנית הנתונה, כך שכל מופע של \_ בתבנית מוחלף באות אחת בדיוק. לדוגמא:

```
suggestCompletion(pool, "d_in_") // may return {"drink", "doing"}
suggestCompletion(pool, "re__r_") // may return {"return", "report",
"regard"}
suggestCompletion(pool, "_") // may return {"i", "a"}
```

- ניתן להניח שמילים במערך הקלט "חוקיות" לפי ההגדרה בסעיף הקודם, ושהן נתונות כ- lowercase.
- ניתן להניח שהתבנית מורכבת ממקפים תחתיים ומאותיות a-z קטנות בלבד.
- ניתן להניח ששני הקלטים לפונקציה אינם null.

(II) הוסיפו פונקציה main במחלקה HangmanHelper, אשר מקבלת בשורת הפקודה שני ארגומנטים: הראשון, מסלול לקובץ, והשני, מחרוזת תבנית. התכנית תיעזר ב- WordPool מהסעיף הקודם כדי לייצר מערך של מילים מן הקובץ, ואז תקרא ל- suggestCompletion עם המערך שהתקבל ועם התבנית. התכנית תדפיס את מערך הפלט של הפונקציה, כל מילה בשורה נפרדת. ניתן לבדוק את עצמכם ע"י הרצת התכנית על הקובץ story.txt המצורף לתרגיל והתבניות לפי הדוגמא למעלה.

### 3. [40 נק']

(I) כעת, ניצור מחלקה בשם AnagramHelper, אשר מטרתה לסייע במשחקי אנגרמות, שהן מילים בשינוי סדר האותיות (למשל, כמו במשחק שבץ-נא). עליכם לממש בתוך המחלקה את השירות הבא:

```
public static String suggestSentence(String[] pool, String scrambledWord)
```

שירות זה יקבל כקלט מאגר מילים ומילה "משובשת" המורכבת מאותיות a-z. כפלט, השירות יחזיר משפט של מילים מן המאגר, המורכב בדיוק מאותיות המילה המשובשת (בתוספת רווחים). אם לא קיים משפט כזה, השירות יחזיר null. לדוגמא:

```
suggestSentence(pool, "ehwoksitdr"); // may return "this worked"
suggestSentence(pool, "rrieyaodantrx"); // may return "extraordinary"
suggestSentence(pool, "dinosaur"); // may return "is a round"
```

**הדרכה:** על מנת לחשב את המשפט ביעילות, נייצג תחילה את מילת הקלט וכל מילה במאגר בתור היסטוגרמה של אותיות (כפי שעשיתם בתרגיל 3 סעיף 9). לדוגמא:

```
"aaceefg" -> [2, 0, 1, 0, 2, 1, 1, 0, ..., 0]
"age"      -> [1, 0, 0, 0, 1, 0, 1, 0, ..., 0]
```

נשים לב כי בהינתן היסטוגרמות  $s$  למילה המשובשת ו- $w$  למילה מן המאגר, האנגרמה של המילה המשובשת מכילה את המילה מן המאגר אם "ם ערכי ההיסטוגרמה של כל אות ב- $s$  גדולים או שווים לאלה שב- $w$ , כלומר, לכל  $i, 0 \leq i < 26$ , מתקיים ש:  $s[i] \geq w[i]$ .

בנוסף, נבחין כי אם המילה  $w$  היא אכן חלק מאנגרמה של  $s$  אזי נוכל לקבל בעיה "קטנה" יותר בה יש לחפש אנגרמה עבור מילה משובשת המורכבת מאותיות  $s$  למעט אלו שהופיעו ב- $w$ . בהיסטוגרמה של מילה זו, ערך התא במקום ה- $i$  הוא בדיוק  $s[i] - w[i]$ . לדוגמא, אם השתמשנו ב- "age" כדי לבנות משפט מ- "aaceefg", נקבל:

```
"aaceefg" -> [2, 0, 1, 0, 2, 1, 1, 0, ..., 0]
"age"      -> [1, 0, 0, 0, 1, 0, 1, 0, ..., 0]
-----
"acef"     -> [1, 0, 1, 0, 1, 1, 0, 0, ..., 0]
```

כעת, נוכל להמשיך ולחפש משפט ממילות המאגר המורכב מן האותיות שנותרו.

אם כן, קיבלנו אלגוריתם למציאת המשפט:

- א. נבנה היסטוגרמה  $s$  על מילת הקלט
- ב. נבנה היסטוגרמות עבור כל המילים במאגר
- ג. נעבור על היסטוגרמות המילים במאגר ונחפש מילה המוכלת באנגרמה של מילת הקלט
- ד. אם מצאנו מילה כזאת (עם היסטוגרמה  $w$ )
  1. נחשב את היסטוגרמת הפרש  $s_2$ , לפי  $s_2[i] = s[i] - w[i]$ .
  2. אם היסטוגרמה  $s_2$  מכילה רק אפסים – סיימנו, ונחזיר את המילה שמצאנו כפלט.
  3. אחרת נקרא לאלגוריתם ברקורסיה על  $s_2$ .

4. אם האלגוריתם מצא משפט עבור s2, נצרף אותו אל המילה של w (מופרד ברווח) ונחזיר את המשפט שנוצר.
5. אחרת, נעבור למילה הבאה.
- ה. אם סיימנו לעבור על כל המילים במאגר ולא מצאנו משפט, נחזיר null.

ממשו את השירות suggestSentence לפי האלגוריתם המוצע לעיל. הערות:

- ניתן להניח שמילים במערך הקלט "חוקיות" לפי ההגדרה בסעיף הקודם, וב-lowercase.
- ניתן להניח שהמחרוזת המשובשת מורכבת מאותיות a-z קטנות בלבד.
- ניתן להניח ששני הקלטים לפונקציה אינם null.

(II) הוסיפו פונקצית main במחלקה AnagramHelper, אשר מקבלת בשורת הפקודה שני ארגומנטים: הראשון, מסלול לקובץ, והשני, מחרוזת משובשת. התכנית תיעזר ב- WordPool מהסעיף הראשון כדי לייצר מערך של מילים מן הקובץ, ואז תקרא ל- suggestSentence עם המערך שהתקבל ועם המילה המשובשת, ותדפיס את הפלט שלה.

כמו בסעיפים קודמים, ניתן לבדוק את עצמכם עם story.txt, אך שימו לב כי ייתכן **שקיים יותר ממשפט מתאים אחד** עבור מילה משובשת, ולכן לא בהכרח תקבלו תוצאה זהה לדוגמאות הנתונות למעלה (וזוה בסדר).

**בהצלחה!**