

הוראות לתרגיל 1 – מבוא לחישוב

ראשית יש להוריד את הקבצים [מהקישור הזה](#), ולפתוח את הפרויקט בסביבת העבודה שלכם. מומלץ לבצע זאת ע"י הרצת הפקודה הנ"ל בterminal:

```
git clone https://github.com/MosheOfer1/Intro_2024B_EX1.git
```

בסעיפים הבאים עליכם להשלים את הפונקציות הריקות במחלקה Logic בלבד, כדי ליצור אלגוריתם שפותר את בעיית 8 המלכות על לוח שחמט. יש לפתור זאת באמצעות אלגוריתם רקורסיבי בשיטת הנסיגה. בכל איטרציה יש להציב בכל השורות החוקיות באותה העמודה ולעבור לעמודה הבאה באיטרציה הבאה.

מומלץ להיעזר בקריאה של [בעיית שמונה המלכות בוויקיפדיה](#).

אחרי שמילאתם את הפונקציות, אתם מוזמנים להריץ את הMain ולבדוק האם הכול עובד כמצופה. בכל הצבה של מלכה על הלוח צריך להופיע באדום כמה פתרונות אפשריים יש, כאשר המלכה הזאת נמצאת שם.

סעיף 1 – 20 נקודות:

השלימו את הפונקציה:

```
public static void printBoard(boolean[][] board) { ... }
```

פונקציה זו מקבלת מערך דו ממדי של משתנים בוליאניים ואמורה להדפיס את לוח השחמט, כאשר 1 מייצג ריבוע עם מלכה (true), ו-0 מייצג ריבוע ריק (false). שימו לב כי array[rows][columns].

לפני כל הדפסה של לוח יש להדפיס 8 סולמיות.

דוגמא לפלט:

```
#####  
11100011  
00011000  
11100111  
00000000  
11100111  
00011000  
11100111  
00000000
```

יש לשים לב כי הבדיקה האוטומטית בחלק הזה משווה תו אחרי תו. ההדפסה חייבת להיעשות במדויק כמו שמפורט ואין לשנות או להוסיף אפילו תו אחד. אין לרדת שורות סתם או להדפיס דברים אחרים. לדוגמא אם מישהו הדפיס 7 סולמיות או רווח אחרי הסולמיות, הציון על כל סעיף זה ואחרים יהיה 0.

סעיף 2 – 30 נקודות:

השלימו את הפונקציה:

```
public static boolean checkBoard(int row, int column, boolean[][] board) { ... }
```

פונקציה זו אמורה לבדוק אם ניתן להוסיף מלכה בעמודה ושורה מסוימים, מבלי שתאכל או תיאכל על ידי מלכות אחרות על הלוח. הפונקציה מקבלת מספר שורה, עמודה ואת מצב הלוח ומחזירה true אם ורק אם ניתן להוסיף מלכה במקום המצוין מבלי לאיים על מלכות אחרות (בודקת את השורה, העמודה והאלכסונים).

סעיף 3 – 20 נקודות:

השלימו את הפונקציה:

```
public static void placeQueensRecursive(int column, boolean[][] board) { ... }
```

פונקציה רקורסיבית זו אמורה למצוא את כל הפתרונות האפשריים לבעיית 8 המלכות, בהינתן העמודה בה נעשית ההצבה ומצב הלוח. בכל פעם שנמצא פתרון יש לעלות באחד את המונה הסטטי solutionsCounter ולהדפיס את הלוח (שימוש בסעיף הראשון). **עליכם להתייחס למצב שבו לפני הקריאה הראשונית לפונקציה, הלוח אינו ריק ומוצבת בו מלכה אחת או יותר.**

סעיף 4 – 30 נקודות:

יש לכתוב unit tests ע"י שימוש בJUNIT עבור כל אחד מהסעיפים הקודמים (לפחות 5 טסטים).

זכרו לכתוב קוד איכותי, עם הערות מנחות ובדיקות תקינות. מכיוון שהסעיפים בנויים זה על גבי זה, טעות נגררת תפגע כנראה בציון לסעיפים הבאים. וודאו את עצמכם טוב לפני ההגשה.

הוראות הגשה

יש להגיש קובץ zip עם רק הקבצים: `LogicTest.java` ו `Logic.java` דרך המודל בתיבת ההגשה, הגשה ביחידים בלבד. יש לקרוא לתיקיה הדחוסה בשם של תעודות הזהות של המגיש, למשל:

123456789.zip

בהצלחה