

# תוכנה 1 - אביב 2019/20

## תרגיל מספר 2

### 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 aviv2 יקרא הקובץ aviv2\_hw2.zip). שימו לב שלא מדובר בשם שלכם, אלא במשתמש האוניברסיטאי, איתו אתם מתחברים למודל למשל.
- קובץ ה-zip יכיל:
  1. קובץ פרטים אישיים בשם details.txt המכיל את שמכם ומספר ת.ז. (אפשר לכתוב בכל שפה – עברית או אנגלית. אתם לא נדרשים להיצמד לפורמט ספציפי, רק שהפרטים יהיו ברורים).
  2. קבצי ה-java של התוכניות אותם התבקשתם לממש. יש לשים את תיקיית ה-src ובתוכה כל קבצי הג'אווה של סעיפים התרגיל. הקפידו ששמות הקבצים יהיו זהים לשמות שהתבקשתם ושימו לב שכולם מתחילים במילה Assignment. אין להוסיף קבצים או תיקיות נוספות! הקפידו שכאשר פותחים את הזיפ, מיד רואים את תיקיית ה-src והיא לא נמצאת בתוך עוד תיקיה, למשל תיקיה ששמה זהה לשם ה-zip או הפרויקט (טעות נפוצה).

שימו לב: חשוב מאד להקפיד על פורמט ההגשה.

כלומר אין להגיש קבצי rar, ויש לקרוא לקבצים בדיוק לפי ההנחיות שקיבלתם. כמו כן אין לצרף תיקיות או קבצים נוספים. **אי עמידה בהנחיות ההגשה תגרור הורדה משמעותית בניקוד!** נדגיש שוב: בזיפ יש רק קובץ details.txt, ותיקיית ה-src שבתוכה כל קבצי הג'אווה שהתבקשתם לכתוב בתרגיל.

בנוסף שימו לב: לפני ההגשה, עליכם להריץ כל תכנית על מספר קלטים שונים (ובפרט לפי הדוגמאות המפורטות בתרגיל) כדי לוודא שהיא אכן פועלת כראוי. זכרו שהארגומנטים בשורת הפקודה (כלומר, הארגומנטים של פונק' ה-main) תמיד מתקבלים כמערך מחרוזות. לכן, אם ברצונכם לעבוד עם מספרים או תווים יש לבצע המרה – אפשר למשל בעזרת השיטות שנלמדו בתרגול.

ניתן להניח כי הקלט תקין, למשל, אם נדרשתם לבצע פעולה על הארגומנט הרביעי - שקיים אחד כזה. כמו כן, **הקפידו על שמות המחלקות במדויק. שם מחלקה לא נכון יגרור הורדה בציון התרגיל.**

- 
1. [30 נק'] חימום: בכל אחד מהסעיפים הבאים עליכם לכתוב תוכנית (=מחלקה עם שירות main) המבצעת משימה מסוימת על הארגומנטים משורת הפקודה. שם התוכנית (=המחלקה והקובץ) יהיה כמפורט בנוהל הגשת התרגילים. קרי, עבור סעיף א' בשאלה 1 כתבו תוכנית בשם Assignment02Q01Sec01, עבור סעיף ב' כתבו תוכנית בשם Assignment02Q01Sec02 וכדומה.

1. התכנית קוראת את כל המחרוזות בקלט, בודקת מהו ערך האסקי של התו הראשון במחרוזת, ובמידה והוא מתחלק ב-3 מדפיסה אותו (את התו) בשורה נפרדת למסך. לדוגמה עבור קלט:

Before a C f

התכנית תדפיס את:

B

f

(כיוון שערך האסקי של B הינו 66, של a הינו 97, של C 67 ושל f 102).

**לא** ניתן להניח כי כל הארגומנטים הינם תווים בודדים (כלומר מחרוזת אשר מכילה תו יחיד). במקרה והמחרוזת היא איננה תו יחיד, למשל "ab", יש לחלץ רק את התו הראשון של המחרוזת (וגם להדפיס רק אותו), במקרה הזה a. היעזרו בתירגול בהרצאה ובעבודה העצמית. (10 נק')

2. התכנית קוראת את כל המחרוזות בקלט, עבור כל מחרוזת סוכמת את ערכי האסקי של כל התווים בה, ומדפיסה למסך את הסכום. כל סכום מודפס בשורה נפרדת.

לדוגמה עבור הקלט: bb "hello world" aaa התכנית תדפיס למסך:

291  
1116  
196

- שימו לב כי המחרוזת "hello world" מוקפת ב"ו" ואילו יתר המחרוזות לא - הגרשיים אינם חלק מן המחרוזות, אלא הדרך להבין כי שתי המילים מייצגות מחרוזת אחת.

3. התכנית קוראת את כל המחרוזות בקלט, ממירה אותן ל-intים, ומחשבת עבור כל ארגומנט את שארית החלוקה שלו ב-3. ניתן להיעזר באופרטור % יש להדפיס את מספר הארגומנטים אשר שארית החלוקה שלהם ב-3 הינה אי-זוגית. ניתן להניח כי כל הארגומנטים הינם מחרוזות אשר מייצגות int חוקי.

לדוגמה עבור קלט: "63 25 8" התכנית תדפיס 1.

כיוון ששארית החלוקה של 8 הינה 2, שארית החלוקה של 25 היא 1 ו-63 הינה 0 (התייחסו ל-0 כשארית חלוקה זוגית), ולכן סה"כ יש ארגומנט אחד אשר שארית החלוקה שלו ב-3 הינה אי-זוגית.

דוגמה נוספת: עבור קלט: "42 66 6" התכנית תדפיס 0, כיוון שלאף אחד מבין שלושת הארגומנטים אין שארית החלוקה ב-3 שהינה אי-זוגית. (10 נק')

2. [20 נק'] ממשיכים להתחמם: כתבו תכנית בשם Assignment02Q02 אשר משמשת כמחשבון לחישוב המספר פאי.

התכנית מקבלת בשורת הפקודה מחרוזת אשר מייצגת מספר טבעי (ניתן להניח כי הועברה מחרוזת אחת בלבד אשר אכן מייצגת מספר טבעי חיובי גדול מ-0) ומחשבת את המספר פאי באמצעות הביטוי המתמטי הבא:

$$\pi = 4 \times \left( 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} + \frac{1}{13} - \frac{1}{15} + \dots \right)$$

כאשר הארגומנט שהתכנית קיבלה יסמן כמה איברים סכמנו.

התכנית תדפיס למסך את הערך שהתקבל ואת הערך של המספר PI כפי שניתן על ידי ה-jdk, שלו ניקרא באמצעות הפקודה Math.PI.

לדוגמא עבור הקלט 4, התכנית תחשב את הביטוי:  $4 \times \left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7}\right)$  ותדפיס למסך:

3.141592653589793 2.8952380952380956 (שני הערכים מודפסים באותה השורה, מופרדים ברווח יחיד).

לדוגמא עבור הקלט 100 התכנית תדפיס למסך: 3.141592653589793 3.1315929035585537

לצורך החישוב השתמשו במשתנים מסוג double (לא float)

3. [20 נק'] כתבו תכנית בשם Assignment02Q03 אשר מקבלת כקלט בשורת הפקודה מספר טבעי (יש להניח כי המחזורת אכן מייצגת מספר טבעי גדול או שווה ל-3), ומדפיסה מספר כזה של איברי סדרת פיבונצ'י (כלומר אם המספר הוא 5, נדפיס את חמשת המספרים הראשונים בסדרה) ואת הסכום שלהם. שני האברים הראשונים הינם 1.

לדוגמא, עבור המספר 20 פלט התכנית יהיה (בפורמט הבא):

The first 20 Fibonacci numbers are:

1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765

The sum is: 17710

שימו לב לפורמט ההדפסה: בשורה הראשונה מודפס כמה איברים הצגנו, בשורה נפרדת כל הסדרה מופרדת ברווח יחיד, ובשורה האחרונה את הסכום שלהם. יש להקפיד על פורמט ההדפסה כולל המלל המדויק שמוצג כאן.

דוגמא נוספת עבור הקלט 10:

The first 10 Fibonacci numbers are:

1 1 2 3 5 8 13 21 34 55

The sum is: 143

4. [20 נק'] נתונה פונקציית ה-main הבאה אשר אמורה לייצר מערך עם 20 המספרים הראשוניים הראשונים. המס' הראשוני הראשון 2 נוסף למערך בשלב האתחול. לאחר מכן, הפונקציה עוברת על מספרים אי-זוגיים החל מ-3 ובודקת עבור כל מספר אם הוא ראשוני. כדי לגלות זאת, הפונקציה מנסה לחלק כל מספר בכל המספרים הראשוניים שהתגלו עד כה, עד לגודל השורש של מספר זה (מכיוון שלא ייתכן שהמספר הוא מכפלה של שני שלמים הגדולים מהשורש שלו...).

```

public static void main(String[] args) {
    int[] primes = new int[20];
    int primeAmount = 1;
    int numToCheck = 3;
    primes[0] = 2;
    while (primeAmount < primes.length) {
        boolean isPrime = true;
        double sqrt = Math.sqrt(numToCheck);
        int top = (int) sqrt;
        for (int i = 0; i < primeAmount && primes[i] < top; i++) {
            if (numToCheck % primes[i] == 0) {
                isPrime = false;
                break;
            }
        }
        if (isPrime) {
            primes[primeAmount] = numToCheck;
            primeAmount++;
        }
        numToCheck += 2;
    }
    System.out.println(Arrays.toString(primes));
}

```

למרבה הצער, נפל באג בתכנית, והיא מדפיסה את הפלט

[2, 3, 5, 7, 9, 11, 13, 15, 17, 19, 23, 25, 29, 31, 35, 37, 41, 43, 47, 49]

אשר מכיל כמה מספרים שאינם ראשוניים (מודגשים בצהוב).

עליכם לגלות מדוע. למשל, התכנית מצאה ש-3 הוא מספר ראשוני, אבל לא מצאה ש-9 מתחלק ב-3.

קראו את [המדריך לשימוש ב-debugger של Eclipse](#), פרקים 1-3.

צרו מחלקה בשם Assignment02Q04 והעתיקו אליה את פונקציית ה-main (ייתכן שתצטרכו להוסיף import-ים בעזרת הפקודה ctrl+shift+o).

הריצו את התכנית במצב דיבאג :



ועקבו אחרי שלבי הריצה, עד שתמצאו את הגורם לבעיה.

תקנו את התכנית (יתכן שצריך לתקן במספר מקומות) והגישו את התכנית המתוקנת.

אם הצלחתם, התכנית תדפיס את המערך הנכון:

[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71]

שימו לב כי אין אמנם אין מגבלה על אופי התיקון, אך התיקון הנדרש לא דורש שינוי גדול בקוד.

5. [10 נק'] נתונה פונקציית ה-main אשר אמורה להדפיס לוח שחמט ריק בגודל 7 על 7 (ולא בגודל 8 על 8 כמו לוח שחמט אמיתי) כלומר, על פלט התכנית להראות כך:

```
X O X O X O X
O X O X O X O
X O X O X O X
O X O X O X O
X O X O X O X
O X O X O X O
X O X O X O X
```

```
public class CheckerBoard {

    public static void main (String[] args) {
        int size = 7; // size of the board

        for (int i = 0; i < size; i++){
            for (int j = 0; j < size; j++){
                if ( i+j % 2 == 0 )
                {
                    System.out.print("X ");
                }
                else
                {
                    System.out.print("O ");
                }
            }
            System.out.println();
        }
    }
}
```

למרבה הצער, נפלה טעות בתכנית והיא מדפיסה את הפלט:

```
X O X O X O X
O O O O O O O
O O O O O O O
O O O O O O O
O O O O O O O
O O O O O O O
O O O O O O O
O O O O O O O
```

צרו מחלקה בשם Assignment02Q05 והעתיקו אליה את פונקציית ה-main (ייתכן שתצטרכו להוסיף import-ים בעזרת הפקודה ctrl+shift+o). הריצו את התכנית במצב דיבאג ועקבו אחרי שלבי הריצה, עד שתמצאו את הגורם לבעיה. תקנו את התכנית והגישו את התכנית המתוקנת. אם הצלחתם, התכנית תדפיס את המערך הנכון.

בהצלחה!!