בית ספר אפי ארזי למדעי המחשב

#### מבוא למדעי המחשב

#### מבחן אמצע 2021

* זמן המבחן: שעתיים. לא תהיה הארכה.
* זמן המבחן מוגבל, ויש לעבוד ביעילות. אם נתקעים בסעיף מסוים, מומלץ לעזוב אותו ולרוץ הלאה.
* השימוש בספרים, סיכומי שיעור, שקפים, מחשבים, מחשבונים, ואינטרנט, או כל דבר אחר פרט לדפי המבחן אסור. כל החומר שצריך כדי לענות על שאלות המבחן ניתן במבחן עצמו. אפשר להשתמש במילון (לא ממוחשב).
* **ענו על כל השאלות ע״י כתיבה בעט על דפי המבחן. כיתבו רק בצד הקדמי של כל דף**;  
  הדפים האחוריים לא ייסרקו ולא יבדקו. ניתן להשתמש בדפי טיוטה, שלא יבדקו.
* אפשר לענות על כל שאלה בעברית או באנגלית, לפי בחירתכם.
* אם תרגישו צורך לעשות הנחה מסוימת כדי לענות על שאלה מסוימת, ניתן לעשות זאת,  
  אם ההנחה סבירה ומנוסחת בדיוק ובבהירות.
* אם אתם לא מסוגלים לתת תשובה מלאה לשאלה מסוימת, תנו תשובה חלקית. תשובה נכונה באופן חלקי תקבל ניקוד חלקי.
* אם אתם לא מצליחים מסיבה כלשהי לכתוב את הקוד הנדרש, ניתן לתאר במילים את המימוש שהייתם רוצים לבצע (בעברית או באנגלית). אם התיאור יהיה מדויק ולעניין, תקבלו ניקוד חלקי.
* אם התבקשתם לכתוב תכנית שאמורה לפעול על קלט מסוים, או פונקציה שפועלת על פרמטרים כלשהם, אזי התכנית / פונקציה לא צריכה לבדוק אם הקלט תקין, אלא אם כן נאמר כך בשאלה במפורש.
* אין צורך לתעד את הקוד שתכתבו במבחן, אלא אם כן אתם רוצים לומר לנו משהו על הקוד שכתבתם.
* התכניות שתכתבו תישפטנה, בין היתר, לפי האורך והאלגנטיות שלהן. תוכניות ארוכות או מסורבלות ללא צורך תקבלנה פחות נקודות, אפילו אם הן ממלאות את המשימה שהוגדרה בשאלה.
* לא יורדו נקודות על טעויות סינטקס טריוויאליות. למשל, במקום לכתוב System.out.println(x) אפשר לכתוב println(x).

בהצלחה!

התבוננו בתוכנית הבאה:

public class Mystery {

public static void main(String[] args) {

int x = 3;

int[] y = { 1, 2, 3 };

triple1(x);

triple2(y);

System.out.println(x);

for (int i = 0; i < x; i++) {

System.out.print(y[i] + " ");

}

}

public static void triple1(int a) {

a = 3 \* a;

}

public static void triple2(int[] a) {

for (int i = 0; i < a.length; i++) {

a[i] = 3 \* a[i];

}

}

}

1. (10 נקודות) (א) כיתבו את הפלט שהתוכנית כותבת, (ב) הסבירו את תשובתכם.

Output:

הסבר:

שאלות 5, 4, 3, 2 עוסקות במחלקה Sets, שמתוארת בדף עזר.

הקדישו חמש דקות לקריאת דף העזר, עכשיו.

2. (10 נקודות) הפונקציה הבאה מחזירה קבוצה שהיא החיתוך (intersection) של שלושת הקבוצות הנתונות. כלומר, אוסף האברים שנמצאים גם ב set1, גם ב set2, וגם ב set3. ממשו את הפונקציה. הקוד שלכם יכול להשתמש בכל פונקציה במחלקת Setsשמתוארת בדף העזר, אפילו אם הפונקציה לא ממומשת.

/\*\* Returns the intersection of the three sets. The intersection of set1, set2, and set3

\* is the set containing the elements that are both in set1, set2, and set3. \*/

public static int[] intersection (int[] set1, int[] set2, int[] set3) {

// Write your code here:

3. (5 נקודות) במחלקת Sets יש שתי פונקציות שנקראות באותו שם (intersection).

(א) איך הקומפיילר יודע לאיזה פונקציה לקרוא?

(ב) איך קוראים לטכניקת התיכנות הזאת?

(ג) האם זאת טכניקת תיכנות מומלצת? אם כן, למה כן? אם לא, למה לא?

4. (15 נקודות) הפונקציה הבאה בודקת אם הקבוצה set1 שולטת בקבוצה set2. כלומר, אם כל איבר של set1 הוא גדול ממש מכל איבר של set2. ממשו הפונקציה. הקוד שלכם יכול להשתמש בכל פונקציה במחלקת Sets שמתוארת בדף העזר, אפילו אם הפונקציה לא ממומשת. טיפ: אלגוריתם יעיל לבעיה הזאת עובר פעם אחת על כל האברים של אחת הקבוצה, ועושה משהו, ופעם אחת על כל האברים של הקבוצה השנייה, ועושה משהו. נסו לממש כזה אלגוריתם.

/\*\* Checks if set1 dominates set2. Set1 dominates set2 if every element of set1

\* is strictly greater (>) than every element of set2. \*/

public static boolean dominates(int[] set1, int[] set2) {

// Write your code here:

5. (20 נקודות) הפונקציה הבאה מחזירה קבוצה שמכילה את כל הערכים האי-זוגיים בקבוצה הנתונה. ממשו את הפונקציה. הקוד שלכם יכול להשתמש בכל פונקציה במחלקת Setsשמתוארת בדף העזר, אפילו אם הפונקציה לא ממומשת.

/\*\* Returns a set containing the odd elements of the given set (elements that are not multiples of 2). \*/

public static int[] oddValuesOf(int[] set) {

// Write your code here:

שאלה 6 עוסקת במחלקה Tables, שמתוארת בדף עזר.

הקדישו חמש דקות לקריאת דף העזר, עכשיו.

6. (20 נקודות) הפונקציה הבאה מחזירה מערך חד-מימדי שמכיל את הממוצעים (averages) של הטורים (columns) בטבלה הנתונה (מערך דו-מימדי שבכל השורות שלו יש אותו מספר אברים). ממשו את הפונקציה.

/\*\* Returns a one-dimensional array containing the averages of the columns of the given

\* table (a two-dimensional array in which all rows have the same number of elements). \*/

public static double[] colAverages(int[][] arr) {

// Write your code here:

7. (20 נקודות) בתרגיל HW4 מימשתם פונקציה שמקבלת מחרוזת (string) של תווים (characters) שמייצגים ספרות ומחזירה את המספר המתאים מטיפוס int. למשל, אם הקלט הוא "536", הפונקציה מחזירה את המספר 536. הפונקציה intToString עושה את הפעולה ההפוכה: מקבלת מספר מטיפוס int גדול מאפס, נניח 536, ומחזירה את המחרוזת המתאימה, במקרה הזה "536".  
לדוגמא, הביטוי intToString(73) + intToString(73) ייצר את המחרוזת "7373".

ממשו את הפונקציה. אם אתם חשים צורך בכך, אתם יכולים להתבונן בטבלת ה ASCII שנמצאת בסוף דפי העזר. שימו לב: פיתרון טריביאלי הוא ״לחבר״ את המספר הנתון ל string ריק. אתם צריכים לעשות את הדבר האמיתי: לטפל בכל סיפרה בנפרד, ולבנות את ה string באופן הדרגתי.

/\*\* Returns the string representation of the given integer value. Assumes that the integer is greater than 0. \*/

public static String intToString(int x) {

// Write your code here: