

## מדעי המחשב

### הוראות

- א. משך הבחינה: שעתיים וחצי.
- ב. מבנה השאלון ומפתח ההערכה: בשאלון זה שני פרקים.
- |           |   |        |   |     |        |
|-----------|---|--------|---|-----|--------|
| פרק ראשון | – | (20x2) | – | 40  | נקודות |
| פרק שני   | – | (30x2) | – | 60  | נקודות |
| סך הכול   | – |        |   | 100 | נקודות |
- ג. חומר עזר מותר בשימוש: כל חומר עזר, חוץ ממחשבון שיש בו אפשרות תכנות.
- ד. הוראה מיוחדת:
- את כל התוכניות שיש לכתוב בשפת מחשב כתבו בשפה אחת בלבד – Java או C#.
- הערה: לא יורדו נקודות אם תכתבו בתוכניות אות גדולה במקום אות קטנה או להפך.

יש לכתוב במחברת הבחינה בלבד. יש לרשום "טיוטה" בראש כל עמוד המשמש טיוטה.  
כתיבת טיוטה בדפים שאינם במחברת הבחינה עלולה לגרום לפסילת הבחינה.

השאלות בשאלון זה מנוסחות בלשון רבים, אף על פי כן על כל תלמידה וכל תלמיד להשיב עליהן באופן אישי.

**בהצלחה!**

## השאלות

בשאלון זה שני פרקים.

יש לענות על שאלות משני הפרקים, לפי ההוראות בכל פרק.

הערה: בכל שאלה שנדרשת בה קליטה, אין צורך לבדוק את תקינות הקלט.

לפותרים בשפת Java: בכל שאלה שנדרשת בה קליטה, הניחו שבתוכנית כתובה ההוראה:

```
Scanner scan = new Scanner (System.in);
```

### פרק ראשון (40 נקודות)

ענו על שתיים מן השאלות 1-3 (לכל שאלה – 20 נקודות).

1. לפניכם הפעולה what בשפת Java או What בשפת C#:

Java	C#
<pre>public static int what (int[] arr) {     int y = 0;     int x = arr[0];     for (int i = 0; i &lt; arr.length; i++)     {         y += arr[i];         x = Math.max (x, y);         y = Math.max (y, 0);     }     return x; }</pre>	<pre>public static int What (int[] arr) {     int y = 0;     int x = arr[0];     for (int i = 0 ;i &lt; arr.Length; i++)     {         y += arr[i];         x = Math.Max (x, y);         y = Math.Max (y, 0);     }     return x; }</pre>

נתון המערך arr:

	0	1	2	3	4	5
arr	3	4	-8	10	-6	8

עקבו בעזרת טבלת מעקב אחר הפעולה what(arr) בשפת Java או What(arr) בשפת C#, וכתבו מה הפעולה מחזירה.

בטבלת המעקב יש לכלול עמודה בעבור כל אחד מן המשתנים האלה: x, y, i.

2. ממשו את הפעולה arrange בשפת Java או Arrange בשפת C# :

Java – public static void arrange (int[] arr)

C# – public static void Arrange (int[] arr)

במערך arr שמתקבל יש מספרים חיוביים ומספרים שליליים.

הפעולה מעבירה לתחילת המערך arr את כל המספרים החיוביים (כולל 0), וממקמת אחריהם את כל המספרים השליליים.

הסדר של המספרים החיוביים (בינם לבין עצמם) יישאר באותו הסדר שבו הם היו לפני הפעולה, וגם הסדר של המספרים

השליליים (בינם לבין עצמם) יישאר באותו הסדר שבו הם היו לפני הפעולה.

דוגמה:

בעבור המערך arr שלפניכם:

	0	1	2	3	4	5	6	7
arr	7	-2	-5	61	4	-1	0	33

המערך arr ייראה כך בסיום הפעולה:

	0	1	2	3	4	5	6	7
arr	7	61	4	0	33	-2	-5	-1

הסבר: הפעולה העבירה לתחילת המערך את המספרים החיוביים, ושמרה על הסדר המקורי של המספרים החיוביים ושל

המספרים השליליים.

.3

מספר חיובי – num ייקרא "מספר נרקיסטי" אם מתקיים התנאי שלפניכם:

מעלים כל אחת מן הספרות של num בחזקת כמות הספרות שיש ב־ num (אורך המספר num), מחברים את כל התוצאות, והסכום שמתקבל שווה ל־ num .

דוגמה ל"מספר נרקיסטי": המספר 407, כי כאשר מעלים כל אחת מן הספרות שלו בחזקת 3 (שהיא כמות הספרות

$$\text{במספר) מקבלים } 407 : 4^3 + 0^3 + 7^3 = 64 + 0 + 343 = 407$$

דוגמה למספר שאינו "מספר נרקיסטי": המספר 58, כי כאשר מעלים כל אחת מן הספרות שלו בחזקת 2 (שהיא כמות

$$\text{הספרות במספר) מקבלים } 89 : 5^2 + 8^2 = 25 + 64 = 89 .$$

**א.** כתבו פעולה חיצונית ששמה isNarc בשפת Java או IsNarc בשפת C#, המקבלת מספר מטיפוס שלם – num הגדול מ־ 0. הפעולה מחזירה true אם המספר נרקיסטי, אחרת היא מחזירה false .

**ב.** כתבו פעולה חיצונית ששמה theNarc בשפת Java או TheNarc בשפת C#, המקבלת מספר מטיפוס שלם – n הגדול מ־ 0. הפעולה מדפיסה את כל המספרים הנרקיסטיים שיש מ־ 1 ועד n (כולל).

הערה: יש להשתמש בפעולה שכתבתם בסעיף א.

**פרק שני (60 נקודות)**

ענו על שתיים מן השאלות 4-6 (לכל שאלה – 30 נקודות).

4. למועצה האזורית "עוז" משתייכים כמה יישובים. המועצה מספקת לילדים הגרים ביישובים אלה שירותי הסעה מן הבית אל בית הספר ובחזרה. הסעה מן הבית אל בית הספר נקראת "הסעה הלוך" והסעה בחזרה מבית הספר אל הבית נקראת "הסעה חזור".

נתונה המחלקה **Transport** – הסעה, ולה ארבע תכונות:

- name – שם היישוב שבעבורו מתבצעת ההסעה, מטיפוס מחרוזת
  - toSchool – סוג ההסעה. בעבור "הסעה הלוך" ערך התכונה הוא true, ובעבור "הסעה חזור" הערך הוא false.
  - num – מספר התלמידים בהסעה – מספר מטיפוס שלם בין 1 ל- 50 (כולל)
  - day – מספר מטיפוס שלם בין 1 ל- 6 (כולל), המייצג את היום בשבוע שבו ההסעה מתבצעת (ראשון-שישי בהתאמה)
- הניחו שקיימות פעולות get/Get לתכונות המחלקה.

א. (1) כתבו פעולה בונה במחלקה **Transport**, המקבלת את שם היישוב שבעבורו ההסעה – name, מספר התלמידים בהסעה – num, והיום בשבוע שבו ההסעה מתבצעת – day.

הפעולה תאתחל את תכונות המחלקה. בסוג ההסעה (toSchool) הפעולה תיצור "הסעה הלוך". הניחו שהפרמטרים שמתקבלים תקינים.

(2) כתבו פעולה בונה במחלקה **Transport** המקבלת רק את מספר התלמידים בהסעה – num. הפעולה תאתחל את תכונות המחלקה כך שתתקבל "הסעה חזור" ביישוב Aviv ביום רביעי עם num תלמידים. הערה: אם פרמטר מספר התלמידים (num) אינו תקין, התכונה תקבל 1.

ב. נתון מערך – arr, מטיפוס **Transport**, המכיל את ההסעות שהתבצעו במהלך השבוע. המערך אינו ממוין לפי סדר כלשהו, ואין בו ערכי null.

(1) כתבו פעולה חיצונית ששמה dayReport בשפת Java או DayReport בשפת C#, המקבלת את

המערך arr, מספר מטיפוס שלם – day המציין יום בשבוע, ופרמטר בוליאני – forward.

אם forward הוא true, הפעולה תחזיר את סך מספר התלמידים שנסעו ב"הסעות הלוך" ביום – day.

אחרת (אם forward הוא false), הפעולה תחזיר את סך מספר התלמידים שנסעו ב"הסעות חזור" ביום – day. הניחו שהפרמטר day שמתקבל תקין (בין 1 ל- 6).

(2) כתבו פעולה חיצונית ששמה moreForward בשפת Java או MoreForward בשפת C#, המקבלת

את המערך arr.

הפעולה תדפיס את מספרי הימים בשבוע שבהם מספר התלמידים שנסעו ב"הסעות הלוך" היה גדול יותר ממספר התלמידים שנסעו ב"הסעות חזור".

הערה: אפשר להיעזר בפעולה שכתבתם בתת-סעיף (1).

5. נתונה המחלקה **User** – משתמש ברשת חברתית, ולה שלוש תכונות:
- name – שם המשתמש מטיפוס מחרוזת (ייתכנו כמה אנשים שלהם אותו השם)
  - id – מספר זהות מטיפוס שלם (מספר ייחודי)
  - friends – מערך מטיפוס שלם, ובו מספרי הזהות של החברים של המשתמש (כל מספר מופיע פעם אחת בלבד). גודל המערך הוא בהתאם לכמות החברים של המשתמש. הניחו שקיימות פעולות get/Get לתכונות המחלקה.
- א. כתבו במחלקה **User** פעולה פנימית ששמה `mutual` בשפת `Java` או `Mutual` בשפת `C#`, המקבלת משתמש (`User`) אחר – `other`. הפעולה מחזירה את כמות החברים המשותפים של שני המשתמשים.
- ב. נתונה המחלקה **SocialNetwork** – רשת חברתית, ולה תכונה אחת:
- users – מערך מטיפוס `User`
- המערך אינו ממוין לפי סדר כלשהו, כל משתמש מופיע בו פעם אחת בלבד, ואין בו ערכי `null`.
- כתבו במחלקה **SocialNetwork** פעולה פנימית ששמה `exactOne` בשפת `Java` או `ExactOne` בשפת `C#`, המקבלת משתמש (`User`) – `other`, שאינו מופיע במערך. הפעולה מחזירה `true` אם קיים במערך משתמש אחד לפחות שלו ול-`other` יש בדיוק חבר אחד משותף. אחרת הפעולה מחזירה `false`.
- הערה: אפשר להיעזר בפעולה שכתבתם בסעיף א.

.6

מספר חיובי – num ייקרא "מספר מושלם" אם מתקיים התנאי שלפניכם:

מחברים את כל המחלקים החיוביים והשלמים של num (המספרים שאם מחלקים את num בהם מקבלים תוצאה ללא שארית), חוץ ממנו, והסכום שמתקבל שווה למספר המקורי.

דוגמה למספר מושלם: המספר 6, כי המחלקים שלו הם: 1, 2, 3, והסכום שלהם הוא 6.

דוגמה נוספת למספר מושלם: המספר 28, כי המחלקים שלו הם: 1, 2, 4, 7, 14, והסכום שלהם הוא 28.

דוגמה למספר שאינו מספר מושלם: המספר 8, כי המחלקים שלו הם: 1, 2, 4, והסכום שלהם הוא 7.

א. (1) כתבו פעולה חיצונית ששמה isPerfect בשפת Java או IsPerfect בשפת C#, המקבלת מספר שלם – num

הגדול מ-2. הפעולה מחזירה true אם הוא מספר מושלם, אחרת היא מחזירה false.

(2) כתבו פעולה חיצונית ששמה thePerfects בשפת Java או ThePerfects בשפת C#, המקבלת שני מספרים

שלמים הגדולים מ-2: low, high. הפעולה תדפיס את כל המספרים המושלמים שבין שני מספרים אלה (כולל).

הניחו ש low קטן מ high או שווה לו.

הערה: אפשר להיעזר בפעולה שכתבתם בתת-סעיף (1).

דוגמה: בעבור low=4 ו high=10 הפעולה תדפיס 6.

הסבר: מן המספר 4 ועד 10, רק המספר 6 הוא מספר מושלם (המספר 4 אינו מספר מושלם כי סכום המחלקים

שלו: 1, 2, הוא 3, המספר 5 אינו מספר מושלם כי רק המספר 1 מחלק אותו, וכן הלאה).

ב. יש השערה שלא קיים מספר אי-זוגי שהוא מספר מושלם.

כתבו פעולה חיצונית ששמה noOdd בשפת Java או NoOdd בשפת C#, המקבלת true אם אכן מבין כל המספרים

עד 999,999 (כולל) אין מספר מושלם שהוא אי-זוגי. אחרת היא מחזירה false.

הערה: אפשר להיעזר בפעולה שכתבתם בתת-סעיף א(1).

## בהצלחה!