

## **מדעי המחשב**

### **הוראות**

- א. משך הבחינה: שעתים וחצי.
- ב. מבנה השאלה ומספר הנקודות: בשאלון זה שני פרקים.  
פרק ראשון – 40 נקודות  
פרק שני – 60 נקודות  

---

סך הכל – 100 נקודות
- ג. חומר עזר מותר בשימוש: כל חומר עזר, חוץ מחשבון שיש בו אפשרות תכננות.
- ד. הורה מיוחדת:  
את כל התוכניות שיש לכתוב בשפת מחשב כתבו בשפה אחת בלבד – Java או C#.  
הערה: לא יורדו נקודות אם כתבו בתוכניות אחרות נדולה במקום אותן קטנה או להפּר.

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

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

**בהצלחה!**

## השאלות

**בשאלון זה שני פרקים.**

**יש לענות על שאלות משנהי הפרקים, לפי ההוראות בכל פרק.**

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

לפתרונות בשפת Java : בכל שאלה שנדרשת בה קליטה, הניחו שבתוכנית כתובה הוראה:

```
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>

: 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 , i , y .

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

**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 החוביים (כולל 0), וממקמת את כל המספרים השליליים.

הסדר של המספרים החוביים (בינם לבין עצמם) ישאר באותו הסדר שבו הם היו לפני הפעולה, וגם הסדר של המספרים השליליים (בינם לבין עצמם) ישאר באותו הסדר שבו הם היו לפני הפעולה.

דוגממה:

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

הסביר: הפעולה העבירה לתחילת המערך את המספרים החוביים, ושמירה על הסדר המקורי של המספרים החוביים ושל

המספרים השליליים.

- .3. מספר חיובי – num יקרא ”מספר נרקיסיסטי” אם מתקיים התנאי ש לפניכם:  
 מעלים כל אחת מן הספרות של num בחזקת כמות הספרות שיש ב- num (אורך המספר num),  
 מוחברים את כל התוצאות, והסכום שモתקבל שווה ל- num .
- דוגמה למספר נרקיסיסטי: המספר 407, כי כאשר מעלים כל אחתמן הספרות שלו בחזקת 3 (שהיא כמות הספרות  
 במספר) מקבלים 407 :  $4^3 + 0^3 + 7^3 = 64 + 0 + 343 = 407$
- דוגמה למספר שאיןו נרקיסיסטי: המספר 58, כי כאשר מעלים כל אחתמן הספרות שלו בחזקת 2 (שהיא כמות  
 הספרות במספר) מקבלים 89 :  $5^2 + 8^2 = 25 + 64 = 89$  .
- א. כתבו פעולה חיצונית בשם IsNarc בשפת C# או Java או IsNarc בשפת C# , המקבלת מספר מטיפוס שלם – num  
 הגדלן מ- 0. הפעולה מחזירה true אם המספר נרקיסיסטי, אחרת היא מחזירה false .
- ב. כתבו פעולה חיצונית בשם theNarc בשפת Java או C# או TheNarc בשפת C# , המקבלת מספר מטיפוס שלם – n  
 הגדלן מ- 0. הפעולה מדפיסה את כל המספרים הנרקיסיסטיים שיש מ- 1 ועד n (כולל).  
הערה: יש להשתמש ב פעולה שכתבתם בסעיף א.

## פרק שני (60 נקודות)

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

4. למועצה האזורית "עו"ז" משתייכים כמה יישובים. המועצה מספקת לילדים הגרים ביישובים אלה שירות הסעה מן הבית אל בית הספר ובחזרה. הנסיעה מן הבית אל בית הספר נקראת "הסעה הלוך" והנסיעה בחזרה מבית הספר אל הבית נקראת "הסעה חזרה".

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

- name – שם היישוב שבבورو מתבצעת הנסיעה, מטיפוס מחרוזת .
- toSchool – סוג הנסיעה. בעבור "הסעה הלוך" ערך התוכנה הוא true, ובעבור "הסעה חזרה" הערך הוא false .
- num – מספר התלמידים בהנסיעה – מספר מטיפוס שלם בין 1 ל- 50 (כולל)
- day – מספר מטיפוס שלם בין 1 ל- 6 (כולל), המציין את היום בשבוע שבו הנסיעה מתבצעת (ראשון-שישי בהתאם)

הנימו שקיימות פועלות get/Get לתכונות המחלקה.

- א. (1) כתבו פולה בונה במחלקה **Transport**, המקבלת את שם היישוב שבבورو הנסיעה – name , מספר התלמידים num – מהירה ומייצרת פעלת **Get** בפעולת **toSchool** – day .

הפעולה תתחול את תכונות המחלקה. בסוג הנסיעה (toSchool) הפולה תיצור "הסעה הלוך".

הנימו שהפרמטרים שמתקבלים תקין.

- (2) כתבו פולה בונה במחלקה **Transport** המקבלת רק את מספר התלמידים בהנסיעה – num .  
הפעולה תתחול את תכונות המחלקה כך שתתקבל "הסעה חזרה" ביחסו Aviv ביום רביעי עם num תלמידים. הערה: אם פרמטר מספר התלמידים (num) אינו תקין, התוכנה תקבל 1 .

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

- (1) כתבו פולה חיונית ששם dayReport Java או C# DayReport בשפה Java או C# , המקבלת את המערך arr , מספר מטיפוס שלם – day המציין יום בשבוע, ופרמטר בוליאי – 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 חזרה" ביום – day .  
הנימו שהפרמטר day שמתקבל תיקין (בין 1 ל- 6 ).

- (2) כתבו פולה חיונית ששם moreForward Java או MoreForward בשפה Java או C# , המקבלת את המערך arr .  
הפעולה תדפיס את מספרי הימים בשבוע שבו מספר התלמידים שנסעו ב"הסעות הלוך" היה גדול יותר ממספר התלמידים שנסעו ב"הסעות חזרה".

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

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

.6

מספר חיובי – num יקרא "מספר מושלם" אם מתקיים התנאי ש לפניכם:

מחברים את כל המחלקים החוביים והשלמים של num (המספרים שאם מחלקיים את num בהם מקבלים תוצאה לא שארית), חוץ ממנו, והסכום שmotekbel 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 מושלם: המספר 8, כי המחלקים שלו הם: 1, 2, 4, והסכום שלהם הוא 7.

א. (1) כתבו פעולה חיצונית ששםה `isPerfect` בשפת Java או `IsPerfect` בשפת C#, המקבלת מספר שלם – num.  
הגודל מ- 2. הפעולה מחזירה `true` אם הוא מספר מושלם, אחרת היא מחזירה `false`.

(2) כתבו פעולה חיצונית ששםה `thePerfектs` בשפת Java או `ThePerfектs` בשפת C#, המקבלת שני מספרים שלמים הגדולים מ- 2: low, high. הפעולה תדפיס את כל המספרים המושלמיים שבין שני מספרים אלה (כולל).  
הניחו ש- low קטן מ- high או שווה לו.

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

דוגמה: בעבר `low=4` ו- `high=10` הפעולה תדפיס 6.

הסביר: מן המספר 4 ועד 10, רק המספר 6 הוא מספר מושלם (המספר 4 אינו מספר מושלם כי סכום המחלקים שלו: 1, 2, 3, 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 ששםה `NoOdd` בשפת Java או `NoOdd` בשפת C#, מחזירה `true` אם אכן מבין כל המספרים עד 3,999,999 (כולל) אין מספר מושלם שהוא אי-זוגי. אחרת היא מחזירה `false`.  
הערה: אפשר להיעזר ב פעולה שכתבתם בתת-סעיף (1).

## בצלחה!