

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 קטע תוכנית, הכתוב בשפות Java ו-C#.

Java	C#
<pre>int n = 23; int steps = 0; while (n > 0) { int root = (int)(Math.sqrt(n)); int square = (int)(Math.pow(root, 2)); n = n - square; steps ++; } System.out.println(steps);</pre>	<pre>int n = 23; int steps = 0; while (n > 0) { int root = (int)(Math.Sqrt(n)); int square = (int)(Math.Pow(root, 2)); n = n - square; steps ++; } Console.WriteLine(steps);</pre>

א. עקבו בעזרת טבלת מעקב אחרי ביצוע קטע התוכנית, וכתבו את המספר שקטע התוכנית מדפיס.

בטבלת המעקב יש לכלול עמודה בעבור כל אחד מן המשתנים האלה: square, root, steps, n.

ב. כתבו דוגמה למספר הגדול מ-1 שמוצב ב-n (בתחילת הקטע) שבעבורו קטע התוכנית מדפיס 1 (אין צורך להציג מעקב).

2. הפרמטר `max` הוא מספר שלם חיובי המייצג את מהירות הנסיעה המקסימלית בקמ"ש המותרת לכלי רכב בקטע כביש מסוים.

הפרמטר `min` הוא מספר שלם חיובי המייצג את מהירות הנסיעה המינימלית בקמ"ש המותרת לכלי רכב באותו קטע כביש.

א. כתבו פעולה חיצונית ששמה `getStatus` בשפת `Java` או `GetStatus` בשפת `C#` המקבלת מספר שלם חיובי `speed`,

המייצג את מהירות הנסיעה בקמ"ש של כלי רכב כלשהו באותו קטע כביש, ואת הפרמטרים `max` ו-`min`.

הפעולה תחזיר מספר שלם על פי התנאים האלה:

– אם מהירות הנסיעה של כלי הרכב גבוהה ב-10 קמ"ש לפחות ממהירות הנסיעה המקסימלית המותרת, הפעולה תחזיר 1.

– אם מהירות הנסיעה של כלי הרכב נמוכה ב-10 קמ"ש לפחות ממהירות הנסיעה המינימלית המותרת, הפעולה תחזיר -1.

– אם שני התנאים שלעיל אינם מתקיימים, הפעולה תחזיר 0.

המערך `speeds` הוא מערך מטיפוס שלם המכיל מידע על מהירויות הנסיעה של כלי רכב שונים באותו קטע כביש מסוים

(כל תא במערך מציג את מהירות הנסיעה של כלי רכב אחר). המערך אינו מסודר בסדר מסוים.

ב. כתבו פעולה חיצונית ששמה `getAllStatus` בשפת `Java` או `GetAllStatus` בשפת `C#` המקבלת את המערך `speeds`

ואת הפרמטרים `max` ו-`min`.

הפעולה תחזיר מערך חדש מטיפוס שלם בגודל זהה למערך `speeds`, המציג בכל תא את המספר 1 או -1 או 0

בהתאם לכלי הרכב שנמצא באותו האינדקס במערך `speeds` (על פי התנאים בסעיף א).

דוגמה: בעבור `min = 30`, `max = 70`, והמערך `speeds`:

	0	1	2	3	4	5	6
speeds	80	50	79	90	20	100	25
	↑	↑	↑	↑	↑	↑	↑
	0	1	2	3	4	5	6
	1	0	0	1	-1	1	0

יחזור המערך שלהלן:

הערות: – יש להשתמש בפעולה שכתבתם בסעיף א.

– אין לשנות את המערך `speeds`.

ג. כלי רכב יקבל קנס אם מהירות הנסיעה שלו בקטע כביש זה גבוהה ב-10 קמ"ש לפחות מן המהירות המקסימלית

המותרת, או נמוכה ב-10 קמ"ש לפחות מן המהירות המינימלית המותרת.

כתבו פעולה חיצונית ששמה `mostFined` בשפת `Java` או `MostFined` בשפת `C#`, המקבלת את המערך `speeds`

ואת הפרמטרים `max` ו-`min`.

הפעולה תחזיר `true` אם לפחות חצי מכמות כלי הרכב במערך `speeds` קיבלו קנס, אחרת הפעולה תחזיר `false`.

דוגמה: בעבור המערך הנתון בסעיף ב הפעולה תחזיר `true`, כי מבין שבעת כלי הרכב במערך, ארבעה מקבלים קנס

(כלי הרכב באינדקסים 0, 3, 4, 5).

הערה: יש להשתמש בפעולה שכתבתם בסעיף ב.

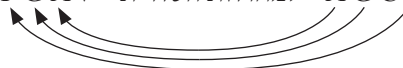
3.

במשחק "בול פגיעה" בוחרים מראש מחרוזת סודית, המורכבת מן האותיות Z–A ללא סדר כלשהו. כל אות יכולה להופיע במחרוזת הסודית פעם אחת לכל היותר (כך שאורך המחרוזת הוא בין 1 ל-26).

המשחק מתנהל באופן שלהלן:

השחקן מנחש מחרוזת מסוימת (שהוא חושב שהיא המחרוזת הסודית). בעבור כל אות במחרוזת הניחוש, הוא מקבל ציון לפי מידת ההצלחה של הניחוש שלו:

- ציון "בול" ניתן בעבור ניחוש נכון של האות ושל המקום שלה במחרוזת הסודית.
 - ציון "פגיעה" ניתן בעבור ניחוש נכון של האות במחרוזת הסודית אך במקום הלא נכון.
 - ציון "החמצה" ניתן בעבור ניחוש של אות שאינה קיימת כלל במחרוזת הסודית.
- לדוגמה: בעבור הניחוש "A G C F" למחרוזת הסודית "F G A V" יקבל השחקן את הציונים שלהלן:



- ציון "בול" אחד בעבור ניחוש נכון של האות G ושל המקום שלה במחרוזת הסודית (באינדקס 1).
- שני ציוני "פגיעה" בעבור ניחוש נכון של האותיות A ו-F במקומות הלא נכונים.
- ציון "החמצה" אחד בעבור ניחוש לא נכון של האות C, שאינה קיימת כלל במחרוזת הסודית.

כתבו פעולה חיצונית ששמה guess בשפת Java או Guess בשפת C#, המקבלת שני פרמטרים:

מחרוזת סודית – secretStr, ומחרוזת ניחוש – playerStr.

הניחו ששתי המחרוזות תקינות (כלומר מופיעות בהן רק האותיות Z–A וכל אות מופיעה בכל מחרוזת פעם אחת לכל היותר).

אם האורך של מחרוזת הניחוש שונה משל המחרוזת הסודית, הפעולה תדפיס הודעת שגיאה "Size Error".

אם האורך של מחרוזת הניחוש זהה לאורך של המחרוזת הסודית, הפעולה תדפיס את מספר ציוני ה"בול", מספר ציוני ה"פגיעה" ומספר ציוני ה"החמצה" בעבור הניחוש של השחקן (כל הדפסה בשורה חדשה).

פרק שני (60 נקודות)

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

4. נתונה המחלקה **Order**, המייצגת פרטי הזמנה של מוצר כלשהו, ולה ארבע תכונות:

- productID – הקוד המזהה של המוצר, מטיפוס שלם בין 0 ל-99 (כולל)
- price – המחיר של המוצר, מטיפוס מספר ממשי
- quantity – כמות היחידות של המוצר, מטיפוס שלם
- delivery – האם כולל משלוח עד הבית, מטיפוס בוליאני (true – בעבור כולל משלוח, false – בעבור לא כולל משלוח)

לקוח שהזמין משלוח עד הבית צריך לשלם **תוספת** של 100 ש"ח. אם הוזמנו 10 יחידות לפחות באותה הזמנה, ונוסף על כך סכום ההזמנה (לא כולל המשלוח) גבוה מ-170.00 ש"ח – המשלוח ניתן **חינם**.

הניחו שקיימות פעולות get/Set ו- set/Set לתכונות המחלקה.

א. כתבו במחלקה **Order** פעולה בונה המקבלת את הקוד המזהה של המוצר – productID ואת מחיר המוצר – price. הפעולה תאתחל את תכונות המחלקה כך שתתקבל הזמנה של המוצר – productID, במחיר של המוצר – price, בכמות של יחידה אחת, עם משלוח עד הבית.

ב. (1) כתבו במחלקה **Order** פעולה פנימית ששמה totalPrice בשפת Java או totalPrice בשפת C#, המחזירה מספר ממשי של הסכום הכולל לתשלום של כל הפריטים בהזמנה **ללא דמי משלוח**.
 (2) כתבו במחלקה **Order** פעולה פנימית ששמה totalCost בשפת Java או totalCost בשפת C#, המחזירה מספר ממשי של הסכום **כולל דמי משלוח** שהלקוח צריך לשלם (אם ההזמנה לא כוללת משלוח עד הבית או המשלוח חינם, יוחזר סכום ההזמנה ללא דמי משלוח).
הערה: יש להשתמש בפעולה שכתבתם בתת-סעיף ב(1).

ג. נתון מערך – allOrders מטיפוס **Order** המכיל הזמנות שבוצעו בשבוע מסוים. המערך אינו מסודר לפי סדר כלשהו ואין בו ערכי null.

כתבו פעולה חיצונית ששמה sumQuantity בשפת Java או SumQuantity בשפת C#, המקבלת את המערך allOrders. הפעולה מחזירה מערך חדש מטיפוס שלם בגודל 100, שבכל תא מופיעה כמות היחידות הכוללת שהוזמנה מן המוצר, שהקוד המזהה שלו זהה לאינדקס של התא.
למשל, אם הוזמנו ממוצר שהקוד המזהה שלו הוא 0 – בהזמנה אחת 15 יחידות, בהזמנה אחרת 20 יחידות, ובהזמנה נוספת 3 יחידות – יופיע בתא של אינדקס 0 המספר 38.

5. נתונה המחלקה **Birthdate** – תאריך לידה, ולה שלוש תכונות:

- day – היום בחודש (מספר בין 1 ל-31), מטיפוס שלם
- month – החודש (מספר בין 1 ל-12), מטיפוס שלם
- year – השנה, מטיפוס שלם (מספר חיובי)

הניחו שקיימות פעולות `get/Set` ו- `set/Set` לתכונות המחלקה.

א. כתבו במחלקה **Birthdate** פעולה פנימית ששמה `sameDate` בשפת Java או `SameDate` בשפת C#

המקבלת עצם אחר – `other` מטיפוס **Birthdate**.

הפעולה מחזירה `true` אם תאריכי הלידה זהים, אחרת היא מחזירה `false`.

בחברת ההייטק "רשת חברתית" פיתחו מערכת שמחפשת דמיון בין לקוחות החברה. לשם כך בנתה החברה את

המחלקה **Client** – לקוח, ולה שלוש תכונות:

- `firstNameInitial` – האות הראשונה בשם הפרטי (אות גדולה באנגלית), מטיפוס `char`
- `lastNameInitial` – האות הראשונה בשם המשפחה (אות גדולה באנגלית), מטיפוס `char`
- `birthdate` – תאריך לידה, מטיפוס **Birthdate**

הניחו שקיימות פעולות `get/Set` ו- `set/Set` לתכונות המחלקה.

ב. (1) ראשי התיבות של שמות שני הלקוחות הם זהים אם גם האות הראשונה בשם הפרטי שלהם זהה וגם האות הראשונה בשם המשפחה שלהם זהה.

כתבו במחלקה **Client** פעולה פנימית ששמה `sameInitials` בשפת Java או `SameInitials` בשפת C#,

המקבלת עצם אחר – `other` מטיפוס **Client**. הפעולה מחזירה `true` אם ראשי התיבות של שמות שני הלקוחות זהים. אחרת הפעולה מחזירה `false`.

(2) במערכת זו, שני לקוחות נחשבים "דומים" אם גם ראשי התיבות של שמותיהם זהים וגם תאריכי הלידה שלהם זהים (אם רק תנאי אחד מתקיים הלקוחות אינם "דומים").

כתבו במחלקה **Client** פעולה פנימית ששמה `areSame` בשפת Java או `AreSame` בשפת C#, המקבלת עצם אחר – `other` מטיפוס **Client**. הפעולה מחזירה `true` אם שני הלקוחות "דומים", אחרת הפעולה מחזירה `false`. יש להשתמש בפעולות שכתבתם בסעיף א ובתת-סעיף ב(1).

ג. במערכת זו, לקוח נחשב "ייחודי" אם אין שום לקוח אחר ש"דומה" לו (כפי שהוגדר בתת-סעיף ב(2)).

כתבו פעולה חיצונית ששמה `uniques` בשפת Java או `Uniques` בשפת C# המקבלת מערך – `clients` מטיפוס **Client** של כל הלקוחות במערכת. המערך אינו מסודר לפי סדר כלשהו ואין בו ערכי `null`. הפעולה תדפיס את האות הראשונה בשם הפרטי של הלקוחות ה"ייחודיים" במערך (כל הדפסה בשורה חדשה). יש להשתמש בפעולה שכתבתם בתת-סעיף ב(2).

6. א. ממשו את הפעולה שלפניכם:

Java – public static int leftDigits (int num, int k)

C# – public static int LeftDigits (int num, int k)

num הוא מספר חיובי, ו- k הוא מספר חיובי, הקטן מכמות הספרות של num (כלומר אורך המספר num).

הפעולה תחזיר את המספר המיוצג על ידי k הספרות שמצד שמאל של המספר num.

דוגמה: עבור $k = 5$ ו- $num = 260043100$, הפעולה תחזיר את המספר 26004.

ב. ממשו את הפעולה שלפניכם:

Java – public static int rightDigits (int num, int k)

C# – public static int RightDigits (int num, int k)

num הוא מספר חיובי, ו- k הוא מספר חיובי, הקטן מכמות הספרות של num (כלומר אורך המספר num).

הפעולה תחזיר את המספר המיוצג על ידי k הספרות שמצד ימין של המספר num (הניחו שהמספר המוחזר תקין

ואינו מתחיל ב-0).

דוגמה: עבור $k = 5$ ו- $num = 260043100$, הפעולה תחזיר את המספר 43100.

ג. מספר שלם חיובי נקרא "קפרקר מאוזן" כאשר הוא מקיים את התנאי שלהלן:

מעלים את המספר בריבוע, את התוצאה מפצלים באמצע לשני חלקים שווים באורכם (כלומר מספר הספרות בתוצאה

הוא זוגי, וכמות הספרות בכל חלק זהה), וסכום שני החלקים שווה למספר המקורי.

שתי דוגמאות: המספרים 9 ו-4950 הם "קפרקר מאוזן".

הסבר:

המספר	העלאה בריבוע	החלק הימני	החלק השמאלי	סכום
9	$9^2 = 81$	1	8	$8 + 1 = 9$
4950	$4950^2 = 24502500$	2500	2450	$2450 + 2500 = 4950$

כתבו פעולה חיצונית ששמה isKapekar בשפת Java או IsKapekar בשפת C#, המקבלת מספר שלם

חיובי - num. הפעולה תחזיר true אם num הוא מספר "קפרקר מאוזן", אחרת הפעולה תחזיר false.

יש להשתמש בפעולות שכתבתם בסעיפים א-ב.

הניחו שאם מעלים את num בריבוע, מספר הספרות בתוצאה הוא זוגי, ושני החלקים של התוצאה תקינים.

בהצלחה!