

שימו לב: בבחינה זו יש הנחיות מיוחדות.
יש לענות על השאלות על פי הנחיות אלה.

מדעי המחשב

הוראות

א. משך הבחינה: שלוש שעות.

ב. מבנה השאלון ומפתח ההערכה: בשאלון זה שני פרקים.

{	פרק ראשון
	פרק שני
(25x4)	
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 ארבע שאלות סך הכול מן הפרקים הראשון והשני (לכל שאלה – 25 נקודות).

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

1. נתון מערך arr, המכיל מספרים שלמים חיוביים.

	0	1	2	3	4	5	6
arr	30	141	324	623	8004	458	6

לפניהם קטע תוכנית, הכתוב בשפות Java ו-C#.

Java	C#
<pre>int count=0; for (int i=0; i< arr.length; i++) { int num=arr[i]/10; int digit=num%10; if (digit==i) count++; }</pre>	<pre>int count=0; for (int i=0; i< arr.Length; i++) { int num=arr[i]/10; int digit=num%10; if (digit==i) count++; }</pre>

א. עקבו בעזרת טבלת מעקב אחר ביצוע קטע התוכנית עבור המערך arr, ורשמו את ערך המשתנה count לאחר ביצוע קטע התוכנית.

בטבלת המעקב יש לכלול עמודה עבור כל אחד מן המשתנים האלה: count, i, num, digit.

ב. כתבו דוגמה למערך arr בגודל 4 המכיל מספרים שלמים חיוביים (גדולים מ־0) שבסיום קטע התוכנית ערכו של המשתנה count יישאר 0.

ג. כתבו דוגמה למערך arr בגודל 4 המכיל מספרים שלמים חיוביים (גדולים מ־0) שבסיום קטע התוכנית ערכו של המשתנה count יהיה שווה ל־4.

2. "מערך סיסמאות" במערכת ממוחשבת של בנק מסוים הוא מערך מטיפוס שלם בגודל 10 שבו הבנק שומר את עשר הסיסמאות האחרונות של הלקוח (כדי לוודא שהלקוח לא יבחר אותן שוב). הסיסמאות נשמרות לפי הסדר מן החדשה ביותר לישנה ביותר. הסיסמה החדשה ביותר (העדכנית) נמצאת בתא 0 והסיסמה הישנה ביותר נמצאת בתא האחרון במערך. כאשר מתקבלת סיסמה חדשה, הסיסמה הישנה ביותר (הסיסמה שבתא האחרון במערך) נמחקת מן המערך, שאר הסיסמאות זזות ימינה כל אחת לתא שלידה, והסיסמה החדשה מופיעה בתא הראשון (אינדקס 0).
- כתבו פעולה בוליאנית חיצונית ששמה `getPass` בשפת Java או `GetPass` בשפת C#, המקבלת מערך סיסמאות מלא – `arr`, וסיסמה – `password`, מטיפוס שלם.
- אם הסיסמה שהתקבלה היא חדשה (כלומר אינה זהה לשום סיסמה שמופיעה במערך), הפעולה תוסיף אותה למערך `arr` בתא 0 (לאחר הזזת שאר הסיסמאות תא אחד ימינה) ותחזיר `true`. אחרת, מערך הסיסמאות יישאר ללא שינוי ויוחזר `false`.

דוגמה: עבור המערך `arr` שלפניכם ו- `password = 300`, המערך יישאר ללא שינוי והפעולה תחזיר `false`.

	0	1	2	3	4	5	6	7	8	9
arr	1223	134	245	300	111	101	777	900	195	1234

הסבר: הסיסמה 300 כבר מופיעה במערך `arr`, באינדקס 3.

דוגמה נוספת: עבור אותו המערך ו- `password = 8888`, הפעולה תחזיר `true`, והמערך ייראה כך בתום הפעולה:

	0	1	2	3	4	5	6	7	8	9
arr	8888	1223	134	245	300	111	101	777	900	195

הסבר: הסיסמה 8888 אינה אחת הסיסמאות שכבר מופיעות במערך.

3. "מערך מעורבל" הוא מערך מטיפוס שלם שהתהליך המתואר לפניכם בוצע בו 30 פעמים:

i. מוגרלים שני מספרי אינדקס תקינים (בגבולות גודל המערך).

ii. הערכים הנמצאים בתאים שמספריהם עלו בהגרלה מתחלפים ביניהם.

הערה: אם בהגרלה עלו שני מספרי אינדקס זהים, ההגרלה לא תיחשב (כלומר היא לא תיספר כחלק מן התהליך

המבוצע 30 פעמים).

כתבו פעולה חיצונית ששמה shuffle בשפת Java או Shuffle בשפת C# המקבלת מערך מטיפוס שלם – arr ומערבלת אותו.

הפעולה תדפיס את ערכי התאים לפי הסדר לאחר הערבול (כל ערך בשורה חדשה).

פרק שני

4. נתונה המחלקה **TourPackage** – חבילת סלולר לטסים לחו"ל, ולה חמש תכונות:

- id – מספר זיהוי של לקוח, מטיפוס שלם
 - price – מחיר החבילה, מטיפוס שלם
 - maxTime – מספר דקות השיחה הכלול בחבילה, מטיפוס שלם
 - maxData – נפח הנתונים (ב־Giga) לגלישה באינטרנט הכלול בחבילה, מטיפוס שלם
 - extra – הסכום לתשלום על חריגה ממספר הדקות ומנפח הנתונים הכלולים בחבילה הבסיסית, מטיפוס שלם
- תעריפי החריגה הם שקל אחד בעבור כל דקת שיחה נוספת, ו־2 שקלים בעבור כל Giga נוסף בנפח הגלישה. הניחו שקיימות פעולות get/Get לתכונות המחלקה.

- א. (1) כתבו פעולה בונה במחלקה **TourPackage**, המקבלת את מספר הזיהוי של לקוח – id, את מחיר החבילה – price, את מספר דקות השיחה הכלול בחבילה – maxTime, ואת נפח הנתונים – maxData. הפעולה מאתחלת את תכונות המחלקה. התכונה extra מאותחלת ל־0.
- (2) כתבו פעולה פנימית במחלקה **TourPackage** ששמה setExtra בשפת Java או SetExtra בשפת C#, המקבלת את מספר דקות השיחה שבהן השתמש הלקוח בפועל בחו"ל – minutes (מטיפוס שלם), ואת נפח הגלישה שבו השתמש הלקוח בפועל בחו"ל – usage (מטיפוס שלם). הפעולה קובעת ערך לתכונה extra בהתאם לחריגה מן החבילה הבסיסית (ככל שיש) ולתעריפי החריגה. הערה: extra היא 0 לפני הפעולה.

- נתון מערך מלא – packages מטיפוס **TourPackage**, שבו נתונים על לקוחות שונים (לאחר שכבר חושב ועודכן הסכום לתשלום עבור חריגה מן החבילה הבסיסית בתכונה extra).
- ב. (1) כתבו פעולה חיצונית ששמה calculate בשפת Java או Calculate בשפת C#, המקבלת את המערך packages, ומחזירה את כמות הלקוחות שצריכים לשלם תוספת על חריגה מן החבילה הבסיסית.
- (2) כתבו פעולה חיצונית ששמה customers בשפת Java או Customers בשפת C#, המקבלת את המערך packages. הפעולה מחזירה מערך מטיפוס שלם שבו מספרי הזיהוי של הלקוחות (id) שצריכים לשלם תוספת על חריגה מן החבילה הבסיסית (גודל המערך הוא לפי כמות האנשים שחרגו). אין חשיבות לסדר ההופעה במערך. הערה: חובה להיעזר בפעולה שכתבתם בסעיף ב(1).

5. נתונה המחלקה Lesson – שיעור, ולה ארבע תכונות:

- id – קוד המקצוע הנלמד, מטיפוס שלם
- hh – השעה שבה השיעור מתחיל (בין 8 ל-17), מטיפוס שלם
- mm – הדקה שבה השיעור מתחיל (בין 0 ל-59), מטיפוס שלם
- duration – משך השיעור בדקות, מטיפוס שלם

הניחו שהפעולות get/Set ו-set/Set קיימות עבור תכונות המחלקה.

הערה: שיעור באותו המקצוע (id) יכול להתקיים כמה פעמים באותו יום לימודים.

נתון מערך שיעורים מלא – arr מטיפוס Lesson של יום לימודים. המערך אינו ממוין.

א. (1) כתבו במחלקה Lesson פעולה פנימית ששמה isLater בשפת Java או IsLater בשפת C#, המקבלת שיעור

אחר – other (מטיפוס Lesson). הפעולה מחזירה true אם השיעור הנוכחי מתחיל מאוחר יותר מן הזמן שמתחיל השיעור האחר (other), ואחרת מחזירה false.

(2) כתבו פעולה חיצונית ששמה last המקבלת את המערך arr, ו**מדפיסה** את קוד המקצוע (id) ששיעורו התחיל אחרון ביום הלימודים.

הניחו שיש רק שיעור אחד המקיים תנאי זה.

הערה: חובה להיעזר בפעולה שכתבתם בסעיף א(1).

ב. (1) כתבו פעולה חיצונית ששמה sumDuration בשפת Java או SumDuration בשפת C# המקבלת את המערך arr וקוד של מקצוע – id. הפעולה מחזירה את מספר הדקות סך הכול של כל השיעורים בקוד המקצוע id המתקיימים באותו יום לימודים.

לדוגמה: אם שיעור במקצוע id נמשך 40 דקות ובאותו היום שיעור נוסף באותו מקצוע נמשך 70 דקות, יוחזר 110.

(2) כתבו פעולה חיצונית ששמה longest בשפת Java או Longest בשפת C# המקבלת את המערך arr. הפעולה תחזיר את קוד המקצוע (id) שמספר הדקות סך הכול של כל שיעוריו באותו יום לימודים הוא הגדול ביותר.

הניחו שיש רק מקצוע אחד המקיים תנאי זה.

הערה: חובה להיעזר בפעולה שכתבתם בסעיף ב(1).

6. נתונה פעולה חיצונית ששמה `digitSum` בשפת Java או `DigitSum` בשפת C#, המקבלת מספר שלם – `num1` ומחזירה

את סכום כל הספרות במספר. אפשר להשתמש בפעולה בלי לממש אותה.

לדוגמה: עבור $num1 = 961$, הפעולה תחזיר 16 ($9 + 6 + 1$).

"סכום הספרות העמוק" של מספר כלשהו הוא מספר חד-ספרתי המתקבל באופן שלהלן:

מחשבים את סכום הספרות שוב ושוב עד שמתקבל מספר חד-ספרתי.

דוגמאות:

"סכום הספרות העמוק" של המספר 5 הוא 5.

"סכום הספרות העמוק" של המספר 36 הוא 9 ($3+6$).

"סכום הספרות העמוק" של המספר 942378 הוא 6, כמפורט להלן:

$$9 + 4 + 2 + 3 + 7 + 8 = 33$$

$$3 + 3 = 6$$

א. (1) כתבו פעולה חיצונית ששמה `deepSum` בשפת Java או `DeepSum` בשפת C# המקבלת מספר שלם `num1`

שאינו שלילי ומחזירה את "סכום הספרות העמוק" שלו.

(2) "סכום הספרות העמוק" יכול להיות אי-זוגי (למשל 5 כמו בדוגמה הראשונה שלעיל) או זוגי (למשל 6 כמו בדוגמה

השלישית שלעיל). יש הטוענים כי בטווח שבין 1 ל-999999 יש יותר מספרים בעלי "סכום ספרות עמוק" זוגי

ממספרים בעלי "סכום ספרות עמוק" אי-זוגי.

כתבו פעולה חיצונית ששמה `isCorrect` בשפת Java או `IsCorrect` בשפת C# המחזירה `true` אם הטענה נכונה,

ואחרת מחזירה `false`.

הערה: חובה להיעזר בפעולה שכתבתם בסעיף א(1).

נתונה פעולה חיצונית נוספת, ששמה `digitExists` בשפת Java או `DigitExists` בשפת C#, המקבלת מספר שלם – `num`

וספרה – `digit`. הפעולה מחזירה `true` אם הספרה `digit` מופיעה במספר `num` לפחות פעם אחת, ואחרת מחזירה `false`.

אפשר להשתמש בפעולה בלי לממש אותה.

ב. כתבו פעולה חיצונית ששמה `inBoth` בשפת Java או `InBoth` בשפת C# המקבלת שני מספרים שלמים: `num1`

ו-`num2`. הפעולה מחזירה `true` אם "סכום הספרות העמוק" של `num1` מופיע במספר `num2`, וגם "סכום הספרות

העמוק" של `num2` מופיע במספר `num1`, אחרת היא מחזירה `false`.

למשל עבור $num1 = 36$ ו- $num2 = 942378$, הפעולה תחזיר `true` כי "סכום הספרות העמוק" של `num1` (9) מופיע

במספר `num2` (942378), ו"סכום הספרות העמוק" של `num2` (6) מופיע במספר `num1` (36).

הערה: חובה להיעזר בפעולה שכתבתם בסעיף א(1).

בהצלחה!