

סוג הבחינה: מבחן מתכונת
מבחן: מתכונת 4
מספר השאלון: 899371
שם התלמיד/ה: _____

מדעי המחשב

יסודות מדעי המחשב – 1-2 יחידות לימוד

הוראות

- א. משך הבחינה: שעה וחצי.
- ב. מבנה השאלון ומפתח ההערכה: בשאלון זה שני פרקים.
פרק ראשון – (2×20) – 40 נקודות
פרק שני – (2×30) – 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);`

השאלות

פרק ראשון (40 נקודות)

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

1. נתון מערך `arr`, המכיל מספרים שלמים:

6	5	4	3	2	1	0	
4	1	7	3	8	2	5	arr

נתון קטע תוכנית, הכתוב בשפות Java ו-C#:

Java

Java

```
int mx = arr[arr.length - 1];
for (int i = arr.length - 2; i >= 0; i--)
{
    if (arr[i] < mx)
        arr[i] = mx - arr[i];
    else
        mx = arr[i];
}
```

#C

C#

```
int mx = arr[arr.Length - 1];
for (int i = arr.Length - 2; i >= 0; i--)
{
    if (arr[i] < mx)
        arr[i] = mx - arr[i];
    else
        mx = arr[i];
}
```

א. עקבו בעזרת טבלת מעקב אחר ביצוע קטע התוכנית, ורשמו את תוכן המערך `arr` לאחר ביצוע קטע התוכנית.

בטבלת המעקב יש לכלול עמודה עבור כל אחד מן המשתנים האלה: `mx`, `arr[i]`, `i`.

ב. כתבו דוגמה למערך `arr` בגודל 5, שלא כל התאים שלו שווים ל-2, ושלאחר ביצוע קטע התוכנית כל התאים שלו יכילו את הערך 2.

2. (20 נקודות)

במועדון שחייה נערך מבחן מהירות. כל שחיין שוחה כמה אורכי בריכה, ולכל אורך נמדד הזמן שלו בשניות (מספר ממשי). שחיין **עובר** את המבחן אם הזמן של כל אחד מהאורכים שלו קטן מ-40 שניות.

הערה: אין צורך לבדוק את תקינות הקלט.

לפותרים בשפת Java: הניחו שבתוכנית כתובה ההוראה:

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

1. כתבו ב־ Java פעולה בשם passTest (ב־ PassTest - C#) שתקבל מספר שלם וגדול מ־ 0 המייצג את מספר האורכים ששחין שחה במבחן. הפעולה תקלוט את הזמן של כל אחד מהאורכים. הפעולה תחזיר true אם השחין עבר את המבחן, אחרת – הפעולה תחזיר false.
 2. כתבו ב־ Java פעולה בשם testDay (ב־ TestDay - C#) שתקלוט בעבור כל אחד מ־ 64 השחיינים שנבחנו ביום מסוים את מספר האורכים שהוא שחה. הפעולה תדפיס את מספר השחיינים שעברו את המבחן ואת מספר השחיינים שלא עברו אותו.
ניתן להשתמש בפעולה שכתבתם בסעיף 1.
 3. (20 נקודות)
במשחק קופסה כל שחקן מחזיק אסימונים. נתון מערך tokens מטיפוס שלם, שבו tokens[i] הוא מספר האסימונים של השחקן שמספרו i. במערך יש לפחות שני תאים, כל הערכים בו אינם שליליים, ולפחות לשחקן אחד יש אסימון.
"סבב מסחר" הוא התהליך המתואר לפניכם, המבוצע 25 פעמים:
i. מוגרלים שני מספרי אינדקס תקינים (בגבולות גודל המערך): הראשון הוא השחקן הנותן, והשני הוא השחקן המקבל.
ii. השחקן הנותן מעביר אסימון אחד לשחקן המקבל.
הערה: אם בהגרלה עלו שני מספרי אינדקס זהים, או שלשחקן הנותן אין אסימונים, ההגרלה לא תיחשב (כלומר היא לא תיספר כחלק מן התהליך המבוצע 25 פעמים).
כתבו פעולה חיצונית ששמה trade בשפת Java או Trade בשפת C# המקבלת את המערך tokens ומבצעת בו "סבב מסחר".
הפעולה תדפיס את מספר האסימונים של כל שחקן לפי הסדר בתום הסבב (כל ערך בשורה חדשה).
כתבו פעולה חיצונית ששמה trade בשפת Java או Trade בשפת C# המקבלת את המערך tokens ומבצעת בו "סבב מסחר".
הפעולה תדפיס את מספר האסימונים של כל שחקן לפי הסדר בתום הסבב (כל ערך בשורה חדשה).
-
- פרק שני (60 נקודות)**
- ענו על **שתיים** מן השאלות 4-6 (לכל שאלה – 30 נקודות).
-
4. (30 נקודות)

בחברת תעופה, כל נוסע רשאי לשלוח מזוודה אחת לבטן המטוס ולעלות למטוס עם תיק יד אחד.

נתונה המחלקה **Bag** - המטען של נוסע, ולה שתי תכונות:

. **weight** - משקל המזוודה הנשלחת לבטן המטוס (בק"ג), מטיפוס שלם

. **handWeight** - משקל תיק היד (בק"ג), מטיפוס שלם

במחלקה **Bag** יש פעולה בונה המקבלת ערכים לשתי התכונות, ופעולות **get/Get** לתכונות.

נתונה גם המחלקה **Passenger** - נוסע בטיסה, ולה ארבע תכונות:

. **ticket** - מספר הכרטיס של הנוסע, מטיפוס שלם

. **allowed** - משקל המזוודה (בק"ג) הכלול במחיר הכרטיס, מטיפוס שלם

. **bag** - המטען של הנוסע, מטיפוס **Bag**

. **fee** - הסכום לתשלום על חריגה במטען, מטיפוס שלם

תעריפי החריגה הם 30 שקלים בעבור כל ק"ג שבו משקל המזוודה עולה על **allowed**, ותשלום קבוע

נוסף של 100 שקלים אם משקל תיק היד גדול מ- 8 ק"ג.

הניחו שקיימות פעולות **get/Get** לתכונות המחלקה **Passenger**.

א. (1) כתבו פעולה בונה במחלקה **Passenger**, המקבלת את מספר הכרטיס - **ticket** ואת משקל

המזוודה הכלול במחיר הכרטיס - **allowed**.

הפעולה מאתחלת את תכונות המחלקה. לנוסע עדיין אין מטען רשום, ולכן התכונה **bag**

מאותחלת ל- **null**, והתכונה **fee** מאותחלת ל- 0.

(2) כתבו פעולה פנימית במחלקה **Passenger** ששמה **checkIn** בשפת **Java** או **CheckIn** בשפת **C#**,

המקבלת את המטען של הנוסע - **b** (מטיפוס **Bag**).

הפעולה שומרת את **b** בתכונה **bag**, וקובעת ערך לתכונה **fee** בהתאם לחריגה במטען (ככל

שיש) ולתעריפי החריגה.

הערה: **fee** היא 0 לפני הפעולה.

ב. (1) נתון מערך מלא - **passengers** מטיפוס **Passenger**, ובו נתונים על נוסעי טיסה אחת (כל

הנוסעים כבר עברו רישום מטען בעזרת הפעולה מסעיף א(2)).

כתבו פעולה חיצונית ששמה **countHeavy** בשפת **Java** או **CountHeavy** בשפת **C#**, המקבלת את

המערך **passengers** ומספר שלם - **limit**. הפעולה מחזירה את מספר הנוסעים שמשקל

המזוודה שלהם גדול מ- **limit**.

(2) כתבו פעולה חיצונית ששמה **heavyTickets** בשפת **Java** או **HeavyTickets** בשפת **C#**, המקבלת

את המערך **passengers** ומספר שלם - **limit**.

הפעולה מחזירה מערך מטיפוס שלם ובו מספרי הכרטיסים (**ticket**) של הנוסעים שמשקל

המזוודה שלהם גדול מ- **limit** (גודל המערך הוא בדיוק כמספר הנוסעים האלה). אין חשיבות

לסדר ההופעה במערך.

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

5. (30 נקודות)

נתונה המחלקה **Playlist** - רשימת השמעה באפליקציית מוזיקה, ולה שלוש תכונות:

· title - שם הרשימה מטיפוס מחרוזת (ייתכנו כמה רשימות שלהן אותו השם)

· code - קוד הרשימה מטיפוס שלם (מספר ייחודי)

· songs - מערך מטיפוס שלם, ובו הקודים של השירים ברשימה (כל קוד מופיע פעם אחת בלבד). גודל המערך הוא בהתאם למספר השירים ברשימה.

הניחו שקיימות פעולות get/Get לתכונות המחלקה.

א. כתבו במחלקה **Playlist** פעולה פנימית ששמה `missingFrom` בשפת `Java` או `MissingFrom` בשפת `C#`, המקבלת רשימת השמעה (**Playlist**) אחרת - `other`. הפעולה **מחזירה** את מספר השירים ברשימה הנוכחית שאינם מופיעים ברשימה `other`.

ב. נתונה המחלקה **MusicLibrary** - ספריית מוזיקה, ולה תכונה אחת:

· lists - מערך מטיפוס **Playlist**

המערך אינו ממורכז לפי סדר כלשהו, כל רשימה מופיעה בו פעם אחת בלבד, ואין בו ערכי `null`.

כתבו במחלקה **MusicLibrary** פעולה פנימית ששמה `countCovering` בשפת `Java` או

`CountCovering` בשפת `C#`, המקבלת רשימת השמעה (**Playlist**) - `p`, שאינה מופיעה במערך.

הפעולה **מחזירה** את מספר הרשימות במערך שמכילות את כל השירים של `p`.

הערה: אפשר להיעזר בפעולה שכתבתם בסעיף א.

6. (30 נקודות)

undefined

א. ממשו את הפעולה שלפניכם:

Java - `public static int digitSum (int num)`

C# - `public static int DigitSum (int num)`

`num` הוא מספר שלם חיובי.

הפעולה תחזיר את סכום הספרות של `num`.

דוגמה: עבור `num = 4735` הפעולה תחזיר את המספר `19 (4 + 7 + 3 + 5)`.

ב. ממשו את הפעולה שלפניכם:

Java - `public static int generate (int num)`

C# - `public static int Generate (int num)`

`num` הוא מספר שלם חיובי.

הפעולה תחזיר את המספר המתקבל מחיבור `num` עם סכום הספרות שלו. נאמר ש-`num` **מייצר** את המספר המוחזר.

דוגמה: עבור `num = 47` הפעולה תחזיר את המספר `58 (47 + 4 + 7)`, כלומר 47 מייצר את 58.

יש להשתמש בפעולה שכתבתם בסעיף א.

ג. מספר שלם חיובי נקרא "מספר עצמאי" אם אין אף מספר שלם חיובי שמייצר אותו (כפי שהוגדר בסעיף ב).

שתי דוגמאות: המספר 20 הוא "מספר עצמאי", והמספר 21 אינו "מספר עצמאי".

הסבר:

המספר	האם יש מספר שמייצר אותו?	מסקנה
21	כן: המספר 15 מייצר אותו ($21 = 5 + 1 + 15$)	אינו מספר עצמאי
20	לא: המספרים 1-9 מייצרים את המספרים הזוגיים 2-18, והמספרים 10-19 מייצרים את המספרים האי-זוגיים 11-29	מספר עצמאי

כתבו פעולה חיצונית ששמה isSelfNumber בשפת Java או IsSelfNumber בשפת C#, המקבלת מספר שלם חיובי - num. הפעולה תחזיר true אם num הוא "מספר עצמאי", אחרת הפעולה תחזיר false.

יש להשתמש בפעולות שכתבתם בסעיפים א-ב.

שימו לב: כל מספר שמייצר את num קטן מ- num.

כל הבגרויות במדעי המחשב עם פתרונות מלאים: www.guyai.co.il/bagrut/computer-science