

סוג הבחינה: מבחן מתכונת
מבחן: מתכונת 3
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. (20 נקודות)

נתונים שני מערכים מטיפוס שלם: a בגודל 8, ו- b בגודל 7.
המערך a מכיל מספרים גדולים מ-0, והמערך b מכיל כולו אפסים.

7	6	5	4	3	2	1	0	
12	12	5	9	2	7	7	4	a

6	5	4	3	2	1	0	
0	0	0	0	0	0	0	b

נתון קטע התכנית שלפניך, הכתוב ב-Java וב-C#.

Java

Java

```
int max = a[0];
int k = 0;
for(int i=1; i<a.length; i++)
{
    if(a[i] > max)
    {
        max = a[i];
        b[k] = i;
        k++;
    }
}
```

#C

C#

```
int max = a[0];
int k = 0;
for(int i=1; i<a.Length; i++)
{
    if(a[i] > max)
    {
        max = a[i];
        b[k] = i;
        k++;
    }
}
```

1. עקבו בעזרת טבלת מעקב אחר ביצוע קטע התכנית בעבור המערכים a ו- b הנתונים, ורישמו את המערך b לאחר ביצוע קטע התכנית.

בטבלת המעקב יש לכלול עמודה לכל אחד מהמשתנים האלה: $b[k]$, k , $\max$, $a[i]$, i .

2. תנו דוגמה מייצגת למערך a בגודל 8 המכיל מספרים גדולים מ-0, שבעבורו, לאחר ביצוע קטע התכנית, בתא b[0] יהיה הערך 7, וכל שאר התאים במערך b יכילו 0.

2. (20 נקודות)

במטבח של מסעדה מוצג על מסך "לוח ההזמנות הפתוחות". הלוח שמור במערך orders מטיפוס שלם בגודל 8. לכל הזמנה יש מספר הזמנה חיובי, ומספרי ההזמנות במערך שונים זה מזה. ההזמנות שמורות לפי סדר הגעתן, החל מתא 0, ואין ביניהן תאים ריקים. תא שאין בו הזמנה מכיל 0, וכל התאים הריקים נמצאים בסוף המערך (ייתכן שאין במערך תאים ריקים כלל). כאשר לקוח מבטל הזמנה, ההזמנה נמחקת מן הלוח: כל ההזמנות שאחריה זזות שמאלה כל אחת לתא שלידה, והתא האחרון במערך מקבל את הערך 0. כתבו פעולה בוליאנית חיצונית ששמה cancelOrder בשפת Java או CancelOrder בשפת C#, המקבלת את המערך orders ומספר הזמנה - num (מספר שלם חיובי). אם ההזמנה num מופיעה במערך, הפעולה תמחק אותה כמתואר ותחזיר true. אחרת, המערך יישאר ללא שינוי ויוחזר false.

נתון המערך orders שלפניכם:

7	6	5	4	3	2	1	0	
0	0	0	130	98	121	117	104	orders

דוגמה: עבור המערך orders ו- num = 121, הפעולה תחזיר true, והמערך ייראה כך בתום הפעולה:

7	6	5	4	3	2	1	0	
0	0	0	0	130	98	117	104	orders

הסבר: ההזמנה 121 נמצאה באינדקס 2. ההזמנות 98 ו-130 (וגם התאים הריקים) זזו תא אחד שמאלה, והתא האחרון קיבל 0.

דוגמה נוספת: עבור אותו המערך המקורי ו- num = 200, המערך יישאר ללא שינוי והפעולה תחזיר false, כי ההזמנה 200 אינה מופיעה בלוח.

כתבו פעולה בוליאנית חיצונית ששמה cancelOrder בשפת Java או CancelOrder בשפת C#, המקבלת את המערך orders ומספר הזמנה - num (מספר שלם חיובי). אם ההזמנה num מופיעה במערך, הפעולה תמחק אותה כמתואר ותחזיר true. אחרת, המערך יישאר ללא שינוי ויוחזר false.

3. (20 נקודות)

מספר שלם חיובי - num ייקרא "מספר הר" אם יש בו לפחות 3 ספרות, וכאשר קוראים את ספרותיו משמאל לימין:

- הספרות **עולות ממש** (כל ספרה גדולה מזו שלפניה) עד ספרה אחת שנקראת "הפסגה", ומיד אחריה
- הספרות **יורדות ממש** (כל ספרה קטנה מזו שלפניה) עד הספרה האחרונה.
- הפסגה אינה הספרה הראשונה במספר ואינה הספרה האחרונה בו.

דוגמאות ל"מספר הר":

- 1352 - הספרות 1, 3, 5 עולות, והספרות 2, 5 יורדות (הפסגה היא 5).
- 12321 - הספרות 1, 2, 3 עולות, והספרות 2, 1 יורדות (הפסגה היא 3).
- 120 - הספרות 1, 2 עולות, והספרות 2, 0 יורדות (הפסגה היא 2).

דוגמאות למספרים שאינם "מספר הר":

- 1233 - יש שתי ספרות סמוכות שוות (3, 3).
- 5310 - הספרות רק יורדות, אין עלייה.
- 1357 - הספרות רק עולות, אין ירידה.
- 1323 - אחרי הירידה מ-3 ל-2 יש שוב עלייה ל-3.

א. כתבו פעולה חיצונית ששמה isMountain בשפת Java או IsMountain בשפת C#, המקבלת מספר מטיפוס שלם - num הגדול מ-0. הפעולה **מחזירה** true אם המספר הוא "מספר הר", אחרת היא מחזירה false.

ב. כתבו פעולה חיצונית ששמה printMountains בשפת Java או PrintMountains בשפת C#, המקבלת מספר מטיפוס שלם - n הגדול מ-0. הפעולה **מדפיסה** את כל "מספרי ההר" שיש מ-1 ועד n (כולל), ובסוף מדפיסה כמה "מספרי הר" נמצאו.

הערה: יש להשתמש בפעולה שכתבתם בסעיף א.

פרק שני (60 נקודות)

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

4. (30 נקודות)

מוזיאון הטבע מציע לקבוצות מבקרים סיורים מודרכים. כל סיור מתקיים בעברית או באנגלית. נתונה המחלקה **Tour** - סיור, ולה ארבע תכונות:

- guide - שם המדריך של הסיור, מטיפוס מחרוזת
- hebrew - שפת הסיור. בעבור סיור בעברית ערך התכונה הוא true, ובעבור סיור באנגלית הערך הוא false.
- visitors - מספר המבקרים בסיור - מספר מטיפוס שלם בין 1 ל-25 (כולל)
- month - מספר מטיפוס שלם בין 1 ל-12 (כולל), המייצג את החודש שבו התקיים הסיור (ינואר-דצמבר בהתאמה)

הניחו שקיימות פעולות get/Get לתכונות המחלקה.

- א.** (1) כתבו פעולה בונה במחלקה **Tour**, המקבלת את שם המדריך - `guide`, את מספר המבקרים - `visitors`, ואת החודש - `month`.
 הפעולה תאטחל את תכונות המחלקה. בשפת הסיור (`hebrew`) הפעולה תיצור סיור בעברית. הניחו שהפרמטרים שמתקבלים תקינים.
- (2) כתבו פעולה בונה במחלקה **Tour** המקבלת רק את החודש - `month`.
 הפעולה תאטחל את תכונות המחלקה כך שיתקבל סיור באנגלית בהדרכת `Maya`, בחודש `month`, עם קבוצה מלאה של 25 מבקרים.
- הערה:** אם הפרמטר `month` אינו תקין, התכונה תקבל 7 (חודש יולי).
- ב.** (1) נתון מערך `arr`, מטיפוס **Tour**, המכיל את כל הסיורים שהתקיימו במוזיאון במהלך שנה אחת. המערך אינו ממוין לפי סדר כלשהו, ואין בו ערכי `null`.
 כתבו פעולה חיצונית ששמה `avgVisitors` בשפת `Java` או `AvgVisitors` בשפת `C#`, המקבלת את המערך `arr`, מספר מטיפוס שלם - `month` המציין חודש, ופרמטר בוליאני - `hebrew`.
 אם `hebrew` הוא `true`, הפעולה תחזיר את ממוצע מספר המבקרים בסיור, מבין הסיורים בעברית שהתקיימו בחודש `month`.
 אחרת (אם `hebrew` הוא `false`), הפעולה תחזיר את ממוצע מספר המבקרים בסיור, מבין הסיורים באנגלית שהתקיימו בחודש `month`.
 אם לא התקיים אף סיור מתאים, הפעולה תחזיר 0.
 הממוצע הוא מספר ממשי (`double`), ויכול להיות לא שלם.
 הניחו שהפרמטר `month` שמתקבל תקין (בין 1 ל-12).
- (2) כתבו פעולה חיצונית ששמה `onlyHebrew` בשפת `Java` או `OnlyHebrew` בשפת `C#`, המקבלת את המערך `arr`.
 הפעולה תדפיס את מספרי החודשים שבהם התקיים לפחות סיור אחד בעברית, ולא התקיים אף סיור באנגלית.
- הערה:** אפשר להיעזר בפעולה שכתבתם בתת-סעיף ב(1).

5. (30 נקודות)

נתונה המחלקה **Time** - שעה ביום, ולה שתי תכונות:

· `hour` - השעה (מספר בין 0 ל-23), מטיפוס שלם

· `minute` - הדקה (מספר בין 0 ל-59), מטיפוס שלם

הניחו שקיימות פעולות `get/Set` ו-`set/Set` לתכונות המחלקה.
 בשאלה זו כל השעות הן באותו יום.

א. כתבו במחלקה **Time** פעולה פנימית ששמה `minutesBetween` בשפת `Java` או `MinutesBetween`

בשפת `C#` המקבלת עצם אחר - `other` מטיפוס **Time**.

הפעולה מחזירה את מספר הדקות שבין שתי השעות (מספר שאינו שלילי, בלי קשר לשאלה איזו מן השעות מוקדמת יותר).

למשל, בין 8:55 ל-9:02 יש 7 דקות.

ב. (1) בתחנת רכבת מרכזית פיתחו מערכת שבודקת את לוח הזמנים של הרכבות היוצאות מן

התחנה. לשם כך נבנתה המחלקה **Train** - רכבת, ולה שלוש תכונות:

· `line` - האות המזהה את קו הרכבת (אות גדולה באנגלית), מטיפוס `char`

platform - מספר הרציף שממנו יוצאת הרכבת, מטיפוס שלם

departure - שעת היציאה של הרכבת, מטיפוס Time

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

כתבו במחלקה Train פעולה פנימית ששמה samePlatform בשפת Java או SamePlatform

בשפת #C, המקבלת עצם אחר - other מטיפוס Train. הפעולה מחזירה true אם שתי

הרכבות יוצאות מאותו רציף. אחרת הפעולה מחזירה false.

(2) במערכת זו, שתי רכבות נמצאות "בהתנגשות" אם גם הן יוצאות מאותו רציף וגם ההפרש בין

שעות היציאה שלהן קטן מ-4 דקות (אם רק תנאי אחד מתקיים, הרכבות אינן בהתנגשות).

כתבו במחלקה Train פעולה פנימית ששמה tooClose בשפת Java או TooClose בשפת #C,

המקבלת עצם אחר - other מטיפוס Train. הפעולה מחזירה true אם שתי הרכבות נמצאות

"בהתנגשות", אחרת הפעולה מחזירה false.

יש להשתמש בפעולות שכתבתם בסעיף א ובתת-סעיף ב(1).

ג. כתבו פעולה חיצונית ששמה countConflicts בשפת Java או CountConflicts בשפת #C המקבלת

מערך - trains מטיפוס Train של כל הרכבות היוצאות מן התחנה ביום מסוים. המערך אינו מסודר

לפי סדר כלשהו ואין בו ערכי null. הפעולה תחזיר את מספר הזוגות של רכבות שנמצאות

"בהתנגשות" (כל זוג נספר פעם אחת בלבד).

למשל, אם במערך יש שלוש רכבות בלבד, וכולן יוצאות מרציף 1 בשעות 7:00, 7:02 ו-7:05 -

הפעולה תחזיר 2 (הזוג של 7:00 ו-7:02 והזוג של 7:02 ו-7:05; בין 7:00 ל-7:05 יש 5 דקות).

יש להשתמש בפעולה שכתבתם בתת-סעיף ב(2).

6. כדי לחסוך מקום, אפליקציית מסרים מסוימת "מכווצת" כל מסר: כל רצף של תווים זהים הסמוכים זה

לזה מוחלף בתו אחד בלבד.

לדוגמה: לאחר כיווץ המילה bookkeeper מתקבל המסר המכווץ bokeper.

כדי שמקבל המסר יוכל לשחזר את המסר המקורי, מצורף למסר המכווץ "מערך מונים". גודל מערך

המונים הוא כאורך המסר המכווץ, ובכל תא מופיע מספר הפעמים שהתו המתאים במסר המכווץ

הופיע ברצף במסר המקורי.

לדוגמה: עבור המסר המכווץ bokeper (של המסר המקורי bookkeeper), מערך המונים הוא:

6	5	4	3	2	1	0
1	1	1	2	2	2	1

הסבר: התו הראשון במסר המכווץ (b) הופיע במסר המקורי פעם אחת, התו השני במסר המכווץ (o)

הופיע במסר המקורי פעמיים ברצף, התו השלישי במסר המכווץ (k) הופיע במסר המקורי פעמיים

ברצף, וכן הלאה.

א. רשמו את המסר המכווץ ואת מערך המונים עבור המסר Mississippi.

ב. כתבו פעולה חיצונית ששמה expand בשפת Java או Expand בשפת #C, המקבלת מחרוזת מכווצת -

str ומערך מונים - arr. הפעולה תחזיר את המחרוזת המקורית.

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