

סוג הבחינה: מבחן מתכנת  
מבחן: מתכנת 4  
מספר השאלון: 899371  
קובץ: תשובות סופיות

## 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 –  $(2 \times 20)$  – 40 נקודות  
פרק שני –  $(2 \times 30)$  – 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);`

## תשובות סופיות

### פרק ראשון (40 נקודות)

#### שאלה 1

- א.  $arr = \{3, 6, 8, 4, 7, 3, 4\}$
- ב. למשל  $\{0, 2, 0, 0, 2\}$  (הופך ל- $\{2, 2, 2, 2, 2\}$ ). כל מערך שבו  $arr[4] = 2$  וכל שאר התאים הם 0 או 2 (ולא כולם 2) מתאים.

#### שאלה 2

1.  $pass = true$ ; קולטים את כל הזמנים, ואם  $pass = false$  -  $time \geq 40$ ; מחזירים  $pass$ .
2. 64 פעמים: קולטים מספר אורכים; אם  $passed - ++passTest(laps)$ ; בסוף מדפיסים  $passed$  ו-64 -  $passed$ .

#### שאלה 3

לולאת  $while$  עם מונה  $count$  שגדל רק כאשר  $giver \neq receiver$  וגם  $tokens[giver] > 0$  (ואז  $tokens[giver]--$  ו- $tokens[receiver]++$ ); בסוף מדפיסים את המערך, ערך בכל שורה.

### פרק שני (60 נקודות)

#### שאלה 4

- א(1).  $public Passenger(int ticket, int allowed)$  - השמה ל- $ticket$  ול- $allowed$ ,  $bag = null$ ,  $fee = 0$ .
- א(2).  $bag = b$ ;  $fee = (b.getWeight() > allowed ? (b.getWeight() - allowed) \cdot 30 : 0) + (b.getHandWeight() > 8 ? 100 : 0)$ .
- ב(1). מונים את הנוסעים שבהם  $limit > getBag().getWeight()$ .
- ב(2).  $int[] tickets = new int[countHeavy(passengers, limit)]$ ; ממלאים בעזרת אינדקס נפרד  $k$  את  $ticket$  של כל נוסע שמשקל המזוודה שלו גדול מ- $limit$ .

#### שאלה 5

- א. לכל שיר ב- $this.songs$  מחפשים אותו במערך של  $other$  (חיפוש סדרתי); אם לא נמצא -  $count++$ . מחזירים  $count$ .
- ב. עוברים על  $lists$ ; אם  $count - missingFrom(lists[i]) == 0$  -  $count++$ . מחזירים  $count$ .

#### שאלה 6

- א.  $num = num / 10$ ;  $sum += num \% 10$ ; (עד ש- $num == 0$ )
- ב.  $return num + digitSum(num)$
- ג. לכל  $m$  מ-1 עד  $num - 1$ : אם  $generate(m) == num$  -  $return false$ ; בסוף  $return true$

כל הבגרויות במדעי המחשב עם פתרונות מלאים: [www.guyai.co.il/bagrut/computer-science](http://www.guyai.co.il/bagrut/computer-science)