

מבוא למדעי המחשב  
202-1-101-1

סמסטר ב', תשס"ז, 2006/7  
**בחינת מועד ב'**

ד"ר צחי רוזן  
גב' ילנה אנוב  
גב' תמר פנחס

משך הבחינה: שעתיים וחצי  
חומר עזר: **אסור**

בבחינה 4 שאלות. ענו על כולן: הניקוד מסתכם ב-100 נקודות.

בשאלות התכנות החשיבות העליונה בד"כ היא על **נכונות** הקוד, מאידך, יעילות וסגנון חשובים גם הם, ולכן תשובה יעילה ומסוגננת תזכה בציון גבוה יותר מאשר תשובה מסורבלת או ארוכה מדי.

בכל השאלות אתם **רשאים** להשתמש בסעיפים קודמים, אם כי לא בהכרח תזדקקו לכך. אתם **רשאים** לעשות זאת אפילו אם לא עניתם על הסעיפים האמורים.

לבחינה מצורף בסופה **דף תשובות**. רשמו את תשובותיכם אך ורק בדף התשובות במקומות המיועדים. המחברת שקיבלתם היא מחברת טיוטה בלבד, והיא וטופס הבחינה עצמו **לא** יימסרו כלל לבדיקה.

**הקפידו** על כתב יד ברור.

**הקפידו** לרשום את מספר הנבחן על דף התשובות.

**בהצלחה!**

## שאלה 1 (35 נקודות)

בשאלה זו אתם מתבקשים להשלים מחלקה עם כמה שיטות סטטיות המחשבות יחד את הערך האמצעי (median) של מערך של מספרים.

**אסור** להוסיף למחלקה לא שדות ולא שיטות מעבר למה שמוגדר במפורש. במימוש השיטות הניחו שהפרמטרים תקינים ואינם null.

**אסור** בשום שלב למיין, לא מיין מלא ולא מיין חלקי, את איברי המערך. אפשר להניח שאיברי המערך שונים זה מזה.

להלן הגדרת המחלקה:

```
public class Median {
    private static double avg(int[] arr, int left, int right) { ... }
    private static void swap(int[] arr, int p, int q) { ... }
    private static int split(int[] arr, int left, int right) { ... }
    private static void split(int[] arr) { ... }
    private static int median(int[] arr) { ... }
}
```

להלן הסברים לשיטות:

- `double avg(int[] arr, int left, int right)` – מקבלת מערך ושני אינדקסים  $left \leq right$  ומחזירה את ממוצע האיברים בתחום `[left .. right]`. אין צורך להשלים את השיטה.
- `void swap(int[] arr, int p, int q)` – מקבלת מערך ושני אינדקסים `p` ו-`q`, ומחליפה בין האיברים שב-`p` וב-`q`. אין צורך להשלים את השיטה.
- `int split(int[] arr, int left, int right)` – מקבלת מערך ושני אינדקסים  $left \leq right$  ומסדרת את איברי המערך בתחום `[left .. right]` בשתי קבוצות: קבוצה של האברים הקטנים מהממוצע בחלק השמאלי של המערך, וקבוצה של הגדולים או השווים לממוצע בחלק הימני. מחזירה את האינדקס של השמאלי ביותר בקבוצת הגדולים.  
למשל, אם האיברים בקטע הם 5, 8, 66, 12, 4, 1, 32, סידור אפשרי שלהם הוא 5, 4, 1, 12, 8, 66, 32, שכן הממוצע הוא 18.29, האיברים 5, 4, 1, 12, 8 קטנים מהממוצע והאיברים 66, 32 גדולים מהממוצע. האינדקס המוחזר הוא של 66.
- `void split(int[] arr)` – מקבלת מערך ומסדרת את איברי המערך בשתי קבוצות: קבוצה של `arr.length/2` האיברים הקטנים ביותר בחלק השמאלי של המערך, וקבוצה של יתר האברים בחלק הימני של המערך.  
למשל, אם איברי המערך הם 5, 8, 66, 12, 4, 1, 32, סידור אפשרי של המערך הוא 5, 4, 1, 12, 8, 66, 32, שכן  $7/2=3$  האיברים הקטנים הם 5, 4, 1 ויתר האיברים הם 12, 8, 66, 32.
- `int median(int[] arr)` – מקבלת כפרמטר מערך ומחזירה את הערך האמצעי שלו.  
למשל אם איברי המערך הם 5, 8, 66, 12, 4, 1, 32, על הפונקציה להחזיר 8.

## סעיף א' (10 נקודות)

השלימו את השיטה `split(int[] arr, int left, int right)` בדף התשובות.

רמז: מספר האיברים בחלק השמאלי של המערך הגדולים מהמוצע שווה למספר האיברים בחלק הימני של המערך שקטנים מהמוצע.

## סעיף ב' (15 נקודות)

השלימו את השיטה `split(int[] arr)` בדף התשובות.

רמז: חזרו מספר פעמים לפי הצורך על חלוקה של קטעים שונים של המערך באמצעות השיטה של הסעיף הקודם עד לקבלת המבוקש.

## סעיף ג' (10 נקודות)

השלימו את השיטה `median(int[] arr)` בדף התשובות.

## שאלה 2 (15 נקודות)

בשאלה זו אתם מתבקשים להשלים שיטה סטטית רקורסיבית אחת.

אין להוסיף שיטות עזר. במימוש השיטה הניחו שהפרמטרים תקינים ושונים מ-`null`.

פריטה של ערך היא קבוצה של ערכים שסכומם שווה לערך הנפרט. למשל `1,1,2,4,10,10` היא פריטה של 28.

השלימו בדף התשובות את הפונקציה הרקורסיבית הבאה:

```
private static void change(int n, int[] value, int[] count, int min)
```

הפונקציה מקבלת את הפרמטרים הבאים:

- `n` – הערך אותו יש לפרוט. הניחו שהערך לא שלילי.
- `value` – מערך של הערכים היכולים להשתתף בפריטה. הניחו שהמערך לא ריק. הניחו שכל הערכים במערך חיוביים. הניחו שכל ערך מופיע במערך פעם אחת בלבד. הניחו שהמערך מכיל את הערך 1.
- `count` – מערך של מונים לכל אחד מהערכים שבמערך `value` המשמש למניה של מספר הפעמים שהערכים הנ"ל נלקחו בפריטה. למשל אם `value[3] = 10` ו-`count[3] = 4` אז הערך 10 נלקח 4 פעמים בפריטה.
- `min` – אינדקס מינימאלי של הערכים במערך `value` היכולים להשתתף בפריטה. למשל, אם `min = 3`, אז רק `value[3]` עד `value[value.length-1]` יכולים להשתתף בפריטה.

הפונקציה מדפיסה ללא חזרות את כל הפריטות האפשריות של  $n$  באמצעות ערכים מהמערך `.value`.

לדוגמא, התוכנית הבאה:

```
public static void main(String[] args) {
    int[] values = { 10, 5, 2, 1 };
    int[] change = new int[values.length];
    change(6, values, change, 0);
}
```

תדפיס את הפלט הבא:

```
5 1
2 2 2
2 2 1 1
2 1 1 1 1
1 1 1 1 1 1
```

### שאלה 3 (20 נקודות)

בשאלה זו אתם מתבקשים להוסיף שתי שיטות למחלקה שמייצגת רשימה מקושרת.

אסור להוסיף למחלקה לא שדות ולא שיטות מעבר למה שמוכתב או ניתן לכם במפורש. במימוש השיטות הניחו שהפרמטרים תקינים ושונים מ-`null`.

להלן ההגדרה של המחלקה:

```
public class LinkedList {
    public LinkedList (Object head, LinkedList tail) { ... }
    public boolean isEmpty () { ... }
    public Object head () { ... }
    public LinkedList tail () { ... }
}
```

המחלקה מייצגת רשימה מקושרת לא ריקה כזוג  $(head, tail)$ , כאשר `head` הוא האובייקט שבראש הרשימה ו-`tail` הוא רשימת שאר איברי הרשימה כסדרם. הרשימה הריקה מיוצגת ע"י אובייקט מיוחד שאין לו לא `head` ולא `tail`.

למשל, הרשימה  $\langle 1, 3, 2 \rangle$  מיוצגת ע"י הזוג  $(1, \langle 3, 2 \rangle)$ , בעוד שהרשימה  $\langle 5 \rangle$  מיוצגת ע"י הזוג  $(5, \langle \rangle)$ , כאשר הרשימה  $\langle \rangle$  היא הרשימה הריקה שמיוצגת ע"י האובייקט המיוחד.

להלן הסברים לבנאי ולשיטות של המחלקה:

- `public LinkedList (Object head, LinkedList tail)` – בנאי שמקבל כפרמטרים אובייקט `head` ורשימה מקושרת `tail` ובונה את הרשימה  $(head, tail)$ .
- `public boolean isEmpty ()` – שיטה שמחזירה `true` אם עצם הנושא הוא הרשימה הריקה.

- `Object head () public` – אם עצם הנושא הוא רשימה לא ריקה, השיטה מחזירה את ראש הרשימה `head`, אחרת, היא אסורה לקריאה.
- `LinkedList tail () public` – אם עצם הנושא הוא רשימה לא ריקה, השיטה מחזירה את זנב הרשימה `tail`, אחרת, היא אסורה לקריאה.

אין צורך להשלים אף אחת מהשיטות הנ"ל.

### סעיף א' (10 נקודות)

הוסיפו למחלקה, בדף התשובות, את השיטה הבאה:

```
public LinkedList concat (LinkedList list)
```

על השיטה לקבל כפרמטר רשימה מקושרת `list` ולהחזיר רשימה מקושרת שמתקבלת משרשר עצם הנושא ל-`list`.

למשל, אם עצם הנושא הוא הרשימה  $\langle 3, 5, 1 \rangle$  ו-`list` היא הרשימה  $\langle 2, 1 \rangle$  אזי על השיטה להחזיר את הרשימה  $\langle 2, 1, 3, 5, 1 \rangle$ .

ברור שבשרשר של רשימה ריקה לרשימה כלשהי `list` מתקבלת `list` עצמה.

### סעיף ב' (10 נקודות)

הוסיפו למחלקה, בדף התשובות, את השיטה הבאה:

```
public LinkedList InsAfter (Object x, LinkedList list)
```

על השיטה לקבל כפרמטר אובייקט `x` ורשימה מקושרת `list` ולהחזיר רשימה מקושרת שמתקבלת מהרשימה של עצם הנושא שלתוכה מכניסים את `list` מייד אחרי המופע הראשון של `x` (כאשר זהות בין אובייקטים מוגדרת ע"י `equals`). אם `x` אינו מופיע ברשימה של עצם הנושא, על הפונקציה להחזיר `null`.

למשל, אם עצם הנושא הוא הרשימה  $\langle 2, 4, 1 \rangle$ , `x=2`, ו-`list` היא הרשימה  $\langle 3, 1 \rangle$ , על השיטה להחזיר את הרשימה  $\langle 2, 3, 1, 4, 1 \rangle$ . אם לעומת זאת `x=5`, על השיטה להחזיר `null`.

## שאלה 4 (30 נקודות)

שאלה זו עוסקת בעצמים. בשאלה אתם מתבקשים להשלים שתי מחלקות הקשורות זו בזו, כפי שהן מוגדרת בהמשך.

אסור להוסיף לממשקים הציבוריים של המחלקות הנ"ל דבר מעבר למה שמוכתב במפורש. לעומת זאת, לצורך מילוי המשימות, אתם רשאים להוסיף למחלקות שדות ושיטות פרטיים.

כרצונכם, להוציא הסתייגות אחת שתפורט בהמשך. במימוש השיטות הניחו שהפרמטרים תקינים ושונים מ-null.

להלן הגדרה של הממשק Queue (תור) כפי שהוגדר בכיתה:

```
public interface Queue {
    void enqueue (Object x);
    Object dequeue ();
    boolean isEmpty ();
}
```

להלן הגדרה של הממשק Stack (מחסנית) כפי שהוגדר בכיתה:

```
public interface Stack {
    void push (Object x);
    Object pop ();
    boolean isEmpty ();
}
```

להלן הגדרה של ממשק בשם Buffer (חוצץ):

```
public interface Buffer {
    void put (Object x);
    Object get ();
    void back ();
}
```

הממשק מגדיר מאגר נתונים (container) בדומה לתור שמתנהג כלהלן:

- `void put (Object x)` - השיטה מוסיפה את האובייקט `x` לסוף מאגר הנתונים. אם `x` הוא `null`, השיטה אינה עושה מאומה.
- `Object get ()` - השיטה מוציאה (מוחקת ומחזירה) את האובייקט שבראש מאגר הנתונים. אם מאגר הנתונים ריק, השיטה מחזירה `null` מבלי להוציא מאומה.
- `void back ()` - השיטה משיבה את האובייקט האחרון שהוצא ממאגר הנתונים וטרם הוחזר אליו למאגר בנתונים. אם אין אובייקט כזה השיטה אינה עושה מאומה.

## סעיף א' (20 נקודות)

השלימו את המחלקה הבאה בדף התשובות:

```
public class CharactersBuffer implements Buffer
```

המחלקה מייצגת חוצץ (buffer) של אובייקטים מסוג `Character` (Character, בדומה ל-Integer, שמהווה מעטפת לטיפוס הפרימיטיבי `int`, מהווה מעטפת לטיפוס הפרימיטיבי `char`).

על מימוש המחלקה להיעשות באמצעות תור ושתי מחסניות בלבד.

לשם כך הוסיפו למחלקה בנאי מהצורה

```
public CharactersBuffer(Queue que, Stack stk1, Stack stk2)
```

שמקבל כפרמטר תור ושתי מחסניות.

מעבר לזה, אין להשתמש במערכים או בכל סוג אחר של אוספי נתונים כמו רשימות מקושרות וכדומה.

### סעיף ב' (10 נקודות)

השלימו בדף התשובות את המחלקה הבאה, בהתאם להגדרות שלמעלה:

```
public class CharacterBufferThatPutStrings extends CharactersBuffer
```

המחלקה מתנהגת כמו `CharactersBuffer`, להוציא את `put` שלא מקבל כפרמטר תו בודד, אלא מחרוזת של תווים מטיפוס `String` ומכניס את כל התווים של המחרוזת כסדרם לחוצץ.

בתשובתכם את רשאים להשתמש בשיטות הבאות של `String`:

○ `public int length()` – מחזירה את האורך של עצם הנושא.

○ `public char charAt(int index)` – מחזירה את התו ה-`index` של עצם הנושא.

או ליצור אובייקט חדש מסוג `Character` באמצעות הבנאי הבא:

○ `public Character(char value)`







שאלה 2 (15 נקודות)

```
private static void change(int n, int[] value, int[] count, int min) {  
    if (_____ ) {  
        _____  
        _____  
        _____  
        _____  
        _____  
    } else  
        _____  
        _____  
        _____  
        _____  
        _____  
        _____  
}
```

שאלה 3 סעיף א' (10 נקודות)

```
public LinkedList concat (LinkedList list) {  
    _____  
    _____  
    _____  
    _____  
    _____  
}
```





