

תאריך הבחינה: 6.2.2011

שמות המרצים: מר שי זקוב  
ד"ר פז כרמי  
פרופ' מייק קודיש  
ד"ר חן קיסר  
ד"ר צחי רוזן

שם הקורס: מבוא למדעי המחשב

מספר הקורס: 202-1-1011

שנה: 2011 סמסטר: א' מועד: ב'

משך הבחינה: 3 שעות

חומר עזר: אסור

### מבחן

אנא קראו את ההוראות שלהלן בעיון:

- במבחן זה 7 שאלות. ענו על כולן.
- בשאלות **שבהן לא מצוין אחרת**, הנכם יכולים לבחור בפתרון רקורסיבי או בפתרון לא רקורסיבי.
- רשמו את תשובותיכם **בדפי התשובות בלבד**. המחברת שקיבלתם היא מחברת טיוטה, והיא לא תימסר כלל לבדיקה. בסיום הבחינה נשמור אך ורק את דפי התשובות. כל שאר החומר יועבר לגריסה.
- החשוב ביותר הוא נכונות הקוד. עם זאת, יעילות, סגנון, וכתובה ברורה חשובים גם הם, ולכן **תשובה לא יעילה או לא ברורה** עלולה לגרום להפחתת נקודות.
- בשאלות התכנות המקום בדף התשובות **מספיק** עבור תשובה המנוסחת כראוי.
- הקפידו על כתב יד ברור. **תשובות מסורבלות או ארוכות מדי לא יזכו בניקוד מלא**. כתבו פתרון לשאלה קודם כול במחברת הטיוטה, ורק לאחר מכן העתיקו פתרון נקי לדף התשובות.
- **אין להוסיף פונקציות עזר** אלא אם כן הדבר נתבקש באופן מפורש בשאלה.
- אין לרשום הערות אך יש לדאוג שהקוד יהיה ברור לחלוטין ללא הערות (ריווח, שמות משתנים, וכד').
- אנא קראו שאלה עד תומה בטרם תענו עליה.
- **הקפידו לרשום בכל דפי התשובות גם את מספר הנבחן ואת מספר החדר שבו אתם נבחנו**.
- אם אינכם יודעים את התשובה לשאלה כלשהי, רשמו "לא יודע/ת" (במקום תשובה) ותזכו ב- 20% מניקוד השאלה. **אין אפשרות לרשום "לא יודע" על סעיף — רק על שאלה שלמה**.

**בהצלחה!**

### שאלה 1 (14 נקודות)

נגדיר סדר על מערכים באופן הבא. בהנתן מערכים  $x$  ו- $y$   $int[]$ , נאמר ש- $x > y$  אם ורק אם המערך  $x$  אינו ריק וגם לכל איבר  $y[j]$  במערך  $y$ , קיים איבר  $x[i]$  במערך  $x$ , כך ש- $x[i] > y[j]$ .

כתבו את הפונקציה `arrayGT` בעלת התבנית הבאה המחזירה `true` אם  $x > y$ , ואחרת מחזירה `false`. ניתן להניח כי המערכים  $x$  ו- $y$  אינם `null`.

```
public static boolean arrayGT(int[] x, int[] y){
    boolean answer;
    // הקוד שלכם כאן
    return answer;
}
```

למשל אם:

```
int[] a = {3,1,2,1};
int[] b = {3};
int[] c = {2,2,2};
int[] d = { };
```

אז

<code>arrayGT(a,b)</code>	<code>→ false</code>	<code>arrayGT(c,a)</code>	<code>→ false</code>
<code>arrayGT(a,c)</code>	<code>→ true</code>	<code>arrayGT(c,b)</code>	<code>→ false</code>
<code>arrayGT(a,d)</code>	<code>→ true</code>	<code>arrayGT(c,d)</code>	<code>→ true</code>
<code>arrayGT(b,a)</code>	<code>→ false</code>	<code>arrayGT(d,a)</code>	<code>→ false</code>
<code>arrayGT(b,c)</code>	<code>→ true</code>	<code>arrayGT(d,b)</code>	<code>→ false</code>
<code>arrayGT(b,d)</code>	<code>→ true</code>	<code>arrayGT(d,c)</code>	<code>→ false</code>

נתון הקוד הבא:

```

public static void sortSederYored(int[] arr) {
    if (arr != null && arr.length > 1){
        int n = arr.length;
        int i = 0;
        while(i < n-1){
            int j = nextIndex(arr, i);
            swap(arr, i, j);
            i = i+1;
        }
    }
}

public static int nextIndex(int[] arr, int from) {
    int ans = from;
    for (int i=from+1; i<arr.length; i=i+1) {
        if (arr[ans] < arr[i])
            ans = i;
    }
    return ans;
}

public static void swap(int[] arr, int i, int j) {
    int tmp = arr[i];
    arr[i] = arr[j];
    arr[j] = tmp;
}

```

ציינו עבור כל אחת מהטענות המופיעות בדף התשובות האם היא טענה נשמרת או לא של לולאת ה-while שבפונקציה `sortSederYored`. (הערה: `[a,b]` מסמן את הערכים שבין `a` ל-`b`, כולל `a` וכולל `b`. שימו לב שאם `a > b` אזי אין ערכים בין `a` ל-`b`.)

שימו לב: בשאלה זו הניקוד עבור סעיף הוא נקודה אחת עבור תשובה נכונה, 0 נקודות עבור אי-סימון תשובה, ותורד חצי נקודה עבור תשובה שגויה. הציון המינימאלי הוא 0 (כלומר לא יינתן בכל מקרה ציון שלילי).

### שאלה 3 (20 נקודות)

נאמר שמחרוזת s1 "שזורה" במחרוזת s2 אם כל האותיות של s1 מופיעות ב s2 ובאותו הסדר.

דוגמאות:

המחרוזת "abc" שזורה במחרוזת "quarsbecy" אך לא במחרוזת "cba".  
המחרוזת "abbc" שזורה במחרוזת "wababec" אך לא במחרוזת "wabec".  
המחרוזת "iinea" שזורה במחרוזת "this is an exam" אך לא במחרוזת "iaina".

כל מחרוזת שזורה בעצמה. המחרוזת הריקה שזורה בכל מחרוזת.

כתבו את הפונקציה **הרקורסיבית** `rec` בעלת התבנית הבאה:

```
public static boolean rec(String s1, String s2){
    boolean answer;
    // הקוד שלכם כאן
    return answer;
}
```

המחזירה `true` אם מחרוזת s1 שזורה במחרוזת s2 ו-`false` אחרת.  
ניתן להניח כי המחרוזות s1 ו- s2 אינן `null`.

#### שאלה 4 (14 נקודות)

הממשק Filter מאפשר לנו לסנן איברים.

```
public interface Filter{
    public boolean accept(Object obj);
}
```

קריאה לשיטה `accept(obj)` תחזיר ערך בוליאני המסמן האם "לקבל" את העצם `obj` (אם הערך המוחזר הוא `true`) או לא.

בעבודה 4 כתבתם את הממשק `Expression` וכן מספר מחלקות המממשות את הממשק.

תזכורת לממשק `Expression`:

תכונה	חתימה
החזרת מחרוזת המתארת את סמל הפעולה האחרונה בביטוי	<code>String getLastOperationSymbol()</code>
שיערוך והחזרת ערכו של הביטוי	<code>double evaluate()</code>
החזרת הנגזרת של הביטוי המתמטי לפי המשתנה <code>var</code>	<code>Expression derivative(Variable var)</code>

הניחו כי נתונה המחלקה `ThresholdFilter` המממשת את הממשק `Filter`. לבנאי במחלקה זו מועבר פרמטר יחיד מטיפוס `double` שהינו ערך סף (`threshold`). השיטה `accept` במחלקה זו "מקבלת" (כלומר מחזירה `true`) רק עצמים מטיפוס `Expression` שערכם גדול או שווה לסף.

דוגמה:

```
Expression expr1, expr2;
// קוד כלשהו היוצר אובייקטים ומפנה אליהם את המשתנים expr1 ו-expr2
System.out.println(expr1.evaluate()); // מדפיס 8
System.out.println(expr2.evaluate()); // מדפיס 1
Filter myFilter = new ThresholdFilter(2);
System.out.println(myFilter.accept(expr1)); // מדפיס true
System.out.println(myFilter.accept(expr2)); // מדפיס false
```

כתבו את המחלקה `DerivativeThresholdFilter` המממשת את הממשק `Filter`. הבנאי במחלקה הינו בעל שני פרמטרים: משתנה מטיפוס `Variable` ומשתנה מטיפוס `ThresholdFilter` (בסדר הזה). על השיטה `accept` במחלקה "לקבל" רק עצמים מטיפוס `Expression` שערך הנגזרת שלהם לפי המשתנה מטיפוס `Variable` שהועבר לבנאי גדול או שווה לסף של הפילטר שהועבר לבנאי. ניתן להניח כי הפרמטרים המועברים לבנאי אינם `null`.

נתון הממשק Iterator (ללא השיטה remove):

```
public interface Iterator {
    boolean hasNext();
    Object next();
}
```

וכן ממשק המגדיר קבוצה כפי שראינו בכיתה, אשר שיטותיו מתנהגות כמפורט בהערות:

```
public interface Set {
    void add(Object obj); // לא עושה כלום, ואם קיים, לא עושה כלום
    void remove(Object obj); // מוציא אובייקט מהקבוצה אם קיים, ואם לא קיים, לא עושה כלום
    boolean contains(Object obj); // בודק האם אובייקט נמצא בקבוצה
    Iterator iterator(); // מחזיר איטרטור על איברי הקבוצה
    Set union(Set other); // קוד שיש להשלים – ראו בהמשך השאלה
}
```

השלימו רק את השיטה union שבמחלקה MySet המממשת את Set. union מחזירה אובייקט מסוג Set שאיבריו הם האיחוד בין הקבוצה המיוצגת במחלקה לקבוצה המועברת כפרמטר. ניתן להניח כי הפרמטר המועבר לשיטה (other) איננו null.

```
public class MySet implements Set {
    public MySet() {
        // קוד של בנאי היוצר קבוצה ריקה שנתון ואינכם צריכים לממש
    }
    // קוד של השיטות האחרות (של הממשק Set) שנתון לכם ואינכם צריכים לממש
    public Set union(Set other) {
        // עליכם להשלים את מימוש שיטה זו בדף התשובות
    }
}
```

תזכורת: איבר נמצא באיחוד של שתי קבוצות אם הוא נמצא בלפחות קבוצה אחת מהשתים.

## שאלה 6 (20 נקודות)

הערה: בשאלה זו ניתן לרשום לא יודע על כל אחד מהסעיפים.

א. (10 נקודות) נתונה המחלקה LinkedList אותה ראיתם בכיתה. להזכירכם:

```
public class LinkedList {  
  
    private Link first;  
    .  
    .  
    .  
    private static class Link {  
  
        private Object data;  
        private Link next;  
        .  
        .  
        .  
    }  
}
```

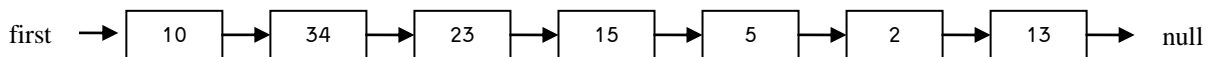
מוסיפים למחלקה LinkedList את השיטה removeOddItems

```
public void removeOddItems() {  
    if (first!=null)  
        first.removeOddItems();  
}
```

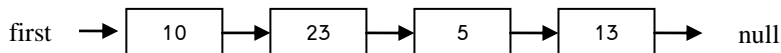
שיטה זו מסירה את האיברים במקומות האי-זוגיים ברשימה. האיבר עליו מצביע first הוא במקום 0 שנחשב מקום זוגי.

עליכם להשלים את השיטה removeOddItems במחלקה Link. אין צורך להשתמש בשיטות של המחלקות הנ"ל.

לדוגמא, כאשר תופעל removeOddItems על הרשימה הבאה:



אזי הרשימה תראה כך בסיום:



ב. (10 נקודות) יובל המבולבל פתר את סעיף (א) אך למרבה הצער אין לסמוך על יובל וייתכן כי הוא גרם ליצירת רשימה מעגלית-פשוטה: זוהי רשימה בה האיבר האחרון מצביע לאיבר הראשון במקום ל-null. על מנת לבדוק אם אכן בידינו רשימה מעגלית-פשוטה או לא נוסף למחלקה LinkedList את השיטה isSimpleCircular המחזירה true אם הרשימה מעגלית-פשוטה, false אחרת:

```
public boolean isSimpleCircular () {  
    boolean ans;  
    if (first==null)  
        ans=false;  
    else  
        ans=first.isSimpleCircular();  
    return ans;  
}
```

עליכם להשלים את השיטה isSimpleCircular במחלקה Link. אין צורך להשתמש בשיטות של המחלקות הנ"ל.

## שאלה 7 (10 נקודות)

השלימו את הפונקציה `findMissingNum` המקבלת כפרמטר מספר שלם  $n > 0$ , קולטת מהמשתמש  $n-1$  מספרים שלמים שונים זה מזה בתחום  $[1..n]$ , ומחזירה את המספר החסר (מהתחום  $[1..n]$ ).

דוגמה: אם קוראים לפונקציה כאשר  $n=6$  ומזינים את חמשת המספרים 4,2,6,1,3, קקלט, אזי על הפונקציה להחזיר את הערך 5.

```
public static int findMissingNum(int n) {
    Scanner sc = new Scanner(System.in);
    int answer;
    int num;
    // קוד שלכם כאן
    for(int i=1; i<n; i=i+1) {
        num = sc.nextInt();
        // עוד קוד שלכם כאן
    }
    // עוד קוד שלכם כאן
    return answer;
}
```

שימו לב: בפתרון שאלה זו אפשר להשתמש במספר כלשהו של משתנים מסוג `int`, אך אין להשתמש במשתנים מסוג אחר. בפרט אין להשתמש במערכים.  
אין צורך לבדוק את נכונות הקלט, וניתן להניח כי  $n > 0$ .