

בוֹחַן ב' - "מבוא למדעי המחשב" 202-1-101-1

סמסטר א' תשס"ב
4.1.2002

פרופ' אורי אברהם
פרופ' דניאל ברנד
ד"ר שמואל ספרוני
פרופ' מיכאל קודיש
ד"ר חן קיסר

משך הבחינה שלוש שעות.
חומר עזר אסור.
אין להשתמש במחשבון.

במבחן זה 7 שאלות.

אנא רשמו את תשובותיכם בדף התשובות בלבד. המחברת שקיבלתם היא **מחברת טיוטה והיא לא תימסר כלל לבדיקה**. בסיום הבחינה נשמור אך ורק את דף התשובות. כל שאר החומר יועבר לגריסה. הקפידו לרשום בדף התשובות גם את מספר הנבחן ומספר החדר שבו אתם נבחנים.

בשאלות התכנות, מספר השורות העומדות לרשותכם בדף התשובות הוא כפול ממספר השורות בתשובות שנפרסם בסוף הבוחן. הקפידו על כתב יד ברור. תשובות מסורבלות או ארוכות מדי לא יזכו בניקוד מלא.

בשאלות 6-7 רשמו את כל התשובות הנכונות.

בהצלחה

שאלה 1 (20 נקודות)

מערך דו-ממדי נקרא מטריצה, אם אינו ריק ואם כל שורותיו שוות באורכן ואורך זה גדול מאפס. השלימו את הגדרת השיטה הסטטית `checkMatrix(int[][] m)`, אשר בודקת אם מערך דו-ממדי `m` הנו מטריצה ומחזירה ערך בוליאני בהתאם. הערה: אם קוראים לשיטה עם פרמטר שערכו `null`, על השיטה להחזיר את הערך `false`. בכל ריבוע ריק יש להשלים הוראה יחידה (כלומר לא יופיע בו הסימן ";").

```
public static boolean checkMatrix( int[][] m){  
  
    boolean tmp =  ;  
  
    for (int i= 0;  ; i = i+1)  
  
        tmp = tmp &  ;  
  
    return tmp;  
}
```

לשם הבהרה, השיטה main הבאה:

```
public static void main (String[] args){  
  
    int[] line1 = {1,2,3,4};  
    int[] line2 = {5,6,7,8};  
    int[] line3 = {9};  
    int[] line4 = new int[0];  
  
    int[][] m1 = {line1,line2};  
    int[][] m2 = {line1,line2,line3};  
    int[][] m3 = null;  
    int[][] m4 = {null,null};  
    int[][] m5 = {line4,line4};  
    int[][] m6 = {line2,null};  
    int[][] m7 = new int[0][0];  
  
    System.out.println(checkMatrix(m1));  
    System.out.println(checkMatrix(m2));  
    System.out.println(checkMatrix(m3));  
    System.out.println(checkMatrix(m4));  
    System.out.println(checkMatrix(m5));  
    System.out.println(checkMatrix(m6));  
    System.out.println(checkMatrix(m7));  
  
}
```

תדפיס:

```
true  
false  
false  
false  
false  
false  
false
```

שאלה 2 (20 נקודות)

נאמר ששני מספרים טבעיים חיוביים הם זרים אם אין להם מחלק משותף (פרט ל-1). נתונה לכם שיטה סטטית `public static int gcd (int m, int n)` אשר מחזירה את המחלק המשותף הגדול ביותר של שני מספרים טבעיים חיוביים. השלימו את השיטה `checkGCD(int[] a)` אשר מחזירה ערך בוליאני המציין האם מערך `a` מקיים אחת משתי התכונות הבאות:

- א. כל שני מספרים שונים ב-`a` זרים,
- ב. כל שני מספרים שונים ב-`a` אינם זרים.

למשל: האוסף `4,7,25` מקיים את תכונה א' והאוסף `6,10,15` מקיים את תכונה ב', למרות שאין שום מספר גדול מ-1 המחלק את שלושת המספרים. האוסף `20,7,20` מקיים את תכונה א' כי רק `7` ו-`20` שונים והם זרים. האוסף `4,7,28` אינו מקיים אף אחת משתי התכונות. הפעלת `checkGCD` על שלושת המערכים הראשונים מחזירה `true` ואילו על הרביעי – `false`.

```
public static boolean checkGCD(int[] a){  
  
    השלם בדף התשובות  
  
}
```

יש להניח שבמערך יש לפחות שני מספרים שונים זה מזה. תוכלו גם להניח שכל המספרים הנתונים הם חיוביים, אך שימו לב שיתכן ואינם שונים זה מזה.

שאלה 3 (20 נקודות)

התבוננו בתוכנית

```
class WhatIsIt {
    public static void main(String [] args) {
        recCheck("", "cab");
    } // main

    static void recCheck(String arg1, String arg2) {
        if (arg2.length() == 1) {
            System.out.println(arg1 + arg2);
            System.out.println(arg1);
        } // if
        else {
            String arg3 = new String(arg1 + arg2.charAt(0));
            String arg4 = new String(arg2.substring(1));
            recCheck(arg3, arg4);
            recCheck(arg1, arg4);
        } // else
    } // recCheck
} // class WhatIsIt
```

א. מה תדפיס התוכנית?

ב. בהנחה שאורך מחזורות הקלט str הוא n. כמה פעמים תיקרא הפונקציה הרקורסיבית recCheck בקריאה ? recCheck("", str)

תזכורות:

1. השיטה charAt(int i) של המחלקה String מחזירה את האות המצויה במקום ה-i של המחזורות. שימו לב שהמקום הראשון במחזורות הוא מקום מספר 0. לדוגמה, השורה System.out.println("ABCD".charAt(1)); תדפיס B.
2. השיטה subString(int i) של המחלקה String מחזירה את התת-מחזורות המתחילה במקום ה-i. לדוגמה, השורה System.out.println("ABCD".substring(1)); תדפיס BCD.

שאלה 4 (15 נקודות)

במחלקה Cover מוגדר מערך `int[] values` כמשתנה מחלקה (סטטי). השלימו את הגדרת השיטה הרקורסיבית `cover(int amount)` אשר בודקת האם ניתן לכסות את הערך `amount` כסכום של חלק (תת-קבוצה) מאיברי המערך `values` ומחזירה ערך בוליאני בהתאם. נגדיר כי תת-הקבוצה הריקה מכסה את הסכום 0 (אפס).

רמז: ניתן להכליל את הבעיה ולהגדיר שיטה `cover(int i, int amount)` אשר בודקת האם ניתן לכסות את הערך `amount` כסכום של חלק מאיברי המערך `values` החל באינדקס `i` (כולל).

```
class Cover{
    public static int[] values = {5,22,13,5,7,-4};
    public static boolean cover(int amount) {
        return cover(0, amount);
    }
    public static boolean cover(int i, int amount){
        השלם בדרך התשובות
    }
} // class Cover
```

לשם הבהרה, השיטה `main` הבאה:

```
public static void main (String[] args){
    System.out.println(cover(42));
    System.out.println(cover(31));
    System.out.println(cover(-5));
    System.out.println(cover(17));
}
```

תדפיס:

```
(42 = 7+13+22 כי true
(31 = 22+13+ (-4 כי true
( כי לא ניתן לכסות את -5 false
(17 = 5+5+7 כי true
```

שאלה 5 (5 נקודות)

תוכנית זו מדפיסה קירוב למספר נתון. מהו?

```
class Why {
    public static void main(String[] args) {
        int n = 0;
        double x,y;
        for (int i = 0; i < 100000; i++) {
            x = Math.random();
            y = Math.random();
            if (x*x + y*y <= 1) n++;
        }
        System.out.println(n/100000);
    }
}
```

שאלה 6 (10 נקודות)

סמנו את כל התשובות הנכונות.

התייחסו לקוד הבא

```
int counter = 0;
while (Math.random() < 2 * Math.random())
    counter = counter + 1;
```

תנאי הלולאה הוא הביטוי המוקף סוגריים בשורה השנייה.

- מכיוון ש $\text{Math.Random}()$ מחזירה ערך חיובי, ומכיוון שכל מספר חיובי קטן מפעמיים אותו מספר, הלולאה לעולם לא תיעצר.
- אם נחליף את תנאי הלולאה ב - $\text{Math.random}() < 1 + \text{Math.random}()$, הלולאה לא תיעצר לעולם.
- אם נחליף את תנאי הלולאה ב - $\text{Math.random}() < 0.5 + \text{Math.random}()$, הלולאה לא תיעצר לעולם.
- אם נהפוך את כוון הסימן < בתנאי ל - > ניתן לצפות שהלולאה תסתיים בתוך מספר סופי של צעדים. יתר על כן, אין סיבה להניח שבממוצע הלולאה תבוצע מספר שונה של פעמים בשני המקרים.
- השינוי המוצע בסעיף הקודם יטה להקטין את מספר הפעמים שהלולאה תבוצע.
 - הקוד דלעיל (זה שלמעלה) והקוד המתקבל על ידי החלפת תנאי הלולאה ב - $\text{Math.random}() < \text{Math.random}() * \text{Math.random}()$ מבצעים שניהם בדיוק אותם חישובים.
 - אף אחת מהתשובות הקודמות אינה נכונה.

שאלה 7 (10 נקודות)

השיטה `public static int someCD(int m, int n)` מחזירה לכל שני שלמים חיוביים m ו- n , מחלק משותף שלהם. המחלק המשותף יהיה גדול מ-1 אלא אם כן 1 הוא המחלק המשותף הגדול ביותר. בכל פעם שתקרא השיטה עם אותם הפרמטרים m ו- n יוחזר אותו מחלק. התבוננו בשיטה הבאה המיועדת למציאת המחלק המשותף הגדול ביותר.

```
Public static int newGCD(int m,int n)
{ return someCD(m,n)*newGCD(m/someCD(m,n), n/someCD(m,n)); }
```

- השיטה מבוססת על נוסחה נכונה ותחזיר תמיד ערך נכון.
- השיטה מבוססת על נוסחה נכונה אך משהו חסר בה ולכן לעולם לא יוחזר ערך נכון.
- השיטה מבוססת על נוסחה מוטעית ותחזיר תמיד ערך מוטעה.
- השיטה מבוססת על נוסחה מוטעית אך במקרים רבים תחזיר ערך נכון.
- הבעיה היחידה עם השיטה היא החלוקה של המספרים השלמים ב - $\text{someCD}(m,n)$. טעויות עלולות לנבוע מחלוק בשלמים.
- השיטה מוטעית משום שהיא קוראת לעצמה, ואי אפשר לחשב משהו באמצעות אותו דבר עצמו.
- אף אחת מהתשובות הקודמות אינה נכונה.

דף תשובות בוחן אמצע – מבוא למדעי המחשב 4.1.2002
 סמנו את כל התשובות הנכונות. בסיום הבוחן נשמור רק את דף התשובות.
 מספר נבחן _____ מספר החזר _____

שאלה מס' 1 checkMatrix (20 נקודות)	השלם כאן את החלקים החסרים
	השלם א'
	השלם ב'
	השלם ג'

שאלה מס' 2 checkGCD (20 נקודות)	שאלה מס' 3 WhatIsIt (15 נק') א' מה תדפיס התוכנית (נק' 15)
	ב' כמה קריאות (5 נק')

שאלה מס' 4 cover (15 נקודות)	שאלה 5 Why (5 נק')

סמנו את כל התשובות הנכונות. (10 נק' כל שאלה)

שאלה 6	א	ב	ג	ד	ה	ו	ז
שאלה 7	א	ב	ג	ד	ה	ו	ז