

סמסטר א' תשס"ו
בוזן ראשון – 2.12.05

דר' ג'יהאד אל-סנע
מר איתי בר-יוסף
גב' דיקלה דותן
מר ענב וינרב
דר' יצחק רוזן

משך הבוחן שעתיים וחצי (לא תינתן הארכה)

חומר עזר - אסור

במבחן זה 7 שאלות בניקוד שונה המסתכם ב-100 נקודות. ענו על כל השאלות.

- בשאלות התכנות, מספר השורות העומדות לרשותכם בדף התשובות רומז על אורך הקוד הנדרש. הקפידו על כתב יד ברור. תשובות מסורבלות או ארוכות מדי לא יזכו בניקוד מלא. אין צורך להעתיק את שורות הקוד הנתונות בשאלון לדף התשובות. מותר להסתמך על סעיפים קודמים גם אם לא עניתם עליהם.
שימו לב: בשאלות התכנות החשיבות העליונה היא לנכונות הקוד. מאידך, יעילות וסגנון חשובים גם הם, ולכן תשובה יעילה ומסוגגנת תזכה בציון גבוה יותר.
- בשאלות בהן יש לסמן את התשובות הנכונות, תתכן יותר מתשובה נכונה אחת. ציון השאלה יקבע לפי היחס בין מספר התשובות הנכונות שסימנתם לבין מספר התשובות הנכונות בסה"כ + מספר התשובות השגויות שסימנתם. למשל, אם בשאלה מסוימת התשובות הנכונות הן א', ג' ו-ד', ונסמן את תשובות א' ו-ג', נקבל $2/3$ מהנקודות, אם נסמן את א', ג', ו-ה', נקבל $2/4$ מהנקודות, ואם נסמן את א', ב', ג', ד', ו-ה' נקבל $3/5$ מהנקודות.
- בשאלות בהן יש לכתוב מהו הקלט שיודפס למסך, תשובה נכונה על סעיף תזכה ב 3 נקודות, תשובה שגויה תוריד 2 נקודות ותשובה שתשאירו ריקה לא תוסיף ולא תוריד נקודות (ניתן לא לענות על כל הסעיפים). לא ניתן לצבור ניקוד שלילי עבור השאלה כולה.
שימו לב: הפלט 3.0 שונה מהפלט 3

בדף האחרון של המבחן ישנה תזכורת למספר פונקציות חשובות, בהן תוכלו להשתמש במהלך המבחן.

אנא רשמו את תשובותיכם **בדפי 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 בשני דפי התשובות גם את מספר הנבחן ומספר החזר שבו אתם נבחנו.

בהצלחה!

שאלה 1 (10 נקודות)

מערך ממוין הוא מערך בו מסודרים כל האיברים בסדר עולה או בסדר יורד.

לדוגמא:

הינם מערכים ממוינים.

3	5	5	7	9
---	---	---	---	---

5	5	0	-1
---	---	---	----

אינו מערך ממוין.

9	2	8	4
---	---	---	---

עליכם להשלים את הפונקציה הבאה, המחזירה ערך אמת אם ורק אם המערך arr הינו ממוין.

יש להקפיד על יעילות מרבית.

הניחו כי arr אינו null.

```
public static boolean isSorted (int [] arr)
{
    boolean increase; // "עולה" = בעברית
    boolean decrease; // " יורד" = בעברית
    boolean ans;
    // השלימו בדף התשובות, שאלה 1
    return ans;
}
```

שאלה 2 (15 נקודות)

מחרוזת הינה פלינדרום אם היא זהה למחרוזת המתקבלת מהפיכת סדר האותיות (קריאת המחרוזת מהסוף להתחלה).

המחרוזות "aiooia" ו-"abcdeba" הינן פלינדרומים, בעוד שהמחרוזות "abcdba" ו-"gdgda" אינן פלינדרומים.

עליכם להשלים את הפונקציה הבאה, המחזירה ערך אמת אם ורק אם המחרוזת s הינה פלינדרום.

המחרוזת הריקה הינה פלינדרום.

הניחו כי s אינה null.

```
public static boolean isPalindrom (String s)
{
    boolean ans;
    // השלימו בדף התשובות, שאלה 2
    return ans;
}
```

תזכורת: ניתן להיעזר בדף האחרון בו רשומות מספר פונקציות חשובות.

שאלה שלישית (15 נקודות)

נתונות הפונקציות הבאות:

```
public static void changeInt (int x)
{
    x = -10;
}

public static void changeArr1 (int [] arr)
{
    int [] tmp = {-10,-20};
    arr = tmp;
}

public static void changeArr2 (int [][] arr, int x)
{
    if (x<arr.length)
    {
        int [] tmp = {-10,-20,-30};
        arr[x] = tmp;
    }
}
```

והפונקציה הראשית הבאה:

```
public static void main (String [] args)
{
    int x=1;
    int [] arr1 = {10,20,30,40};
    int [][] arr2 = {{1,2,3}, {4,5,6}, {7,8}, {1,2,3}};

    System.out.println(arr2[0][0]==x);

    System.out.println(arr2[0]==arr2[3]);

    changeInt(x);
    System.out.println(x);

    changeArr1(arr1);
    System.out.println(arr1[1]);

    changeArr2(arr2, x);
    System.out.println(arr2[0][0]+", "+arr2[x][2]);
}
```

עבור כל אחד מהסעיפים, השלימו בדף התשובות את הפלט שיודפס למסך.

שימו לב: תשובה נכונה לכל סעיף מקנה 3 נקודות. תשובה שגויה מורידה 2 נקודות. שורה ריקה לא מורידה ניקוד. לא יינתן ניקוד שלילי עבור השאלה כולה.

שאלה רביעית (15 נקודות)

נתונות הפונקציות הבאות

```
public static double myDivide (int x, int y)
{
    return ((x/y)+1);
}

public static double myDivide (int x, double y)
{
    return (x/y);
}
```

בנוסף, נתון קטע הקוד הבא:

```
int x=1, y=5, z=0;
double a=2, b=1.0, c=0;
// פה מתווספת שורה נוספת, לפי כל אחד מהסעיפים
```

עבור כל אחד מהסעיפים הבאים, סמנו בדף התשובות האם הוספת הפקודה במקום ההערה בקוד תגרור:

(א) שגיאת קומפילציה.

(ב) שגיאה בזמן ריצה.

(ג) הדפסת פלט למסך, במקרה זה יש לרשום מהו הפלט שיודפס.

שימו לב: תשובה נכונה לכל סעיף מקנה 3 נקודות. תשובה שגויה מורידה 2 נקודות. שורה ריקה לא מורידה ניקוד. לא יינתן ניקוד שלילי עבור השאלה כולה. אין קשר בין הסעיפים.

תזכורת: הפלט 3.0 שונה מהפלט 3

סעיף א'

```
System.out.println(myDivide(x,y));
```

סעיף ב'

```
System.out.println(myDivide(a,(double)y));
```

סעיף ג'

```
System.out.println(myDivide(x,(double)y));
```

סעיף ד'

```
System.out.println(myDivide(5,'D'-'B'));
```

סעיף ה'

```
System.out.println(myDivide(y,z));
```

שאלה חמישית (25 נקודות)

זוג פיתגוראי הוא זוג מספרים שלמים וגדולים מ-0, מהצורה $\langle a, b \rangle$ כך שקיים מספר שלם c המקיים $a^2 + b^2 = c^2$. לדוגמה, $\langle 3, 4 \rangle$ הינו זוג פיתגוראי כי קיים המספר השלם 5 המקיים: $3^2 + 4^2 = 9 + 16 = 25 = 5^2$. בשאלה זו נתייחס לשני זוגות פיתגוראים כאל "זהים" אם הם מהצורה $\langle a, b \rangle$, $\langle b, a \rangle$. למשל, הזוגות $\langle 3, 4 \rangle$ ו- $\langle 4, 3 \rangle$ יקראו זוגות פיתגוראים זהים.

הערה חשובה: ניתן להעזר בסעיפים קודמים גם במידה ולא פתרתם אותם.

סעיף א' (7 נקודות)

השלימו את הפונקציה הסטטית הבאה המקבלת כקלט שני מספרים שלמים וגדולים מ-0 ו- b , ובודקת האם הם יכולים להוות זוג פיתגוראי. אם כן, הפונקציה מחזירה את המספר השלם c כך ש $a^2 + b^2 = c^2$, אחרת הפונקציה מחזירה -1.

מותר להשתמש בפונקציה `Math.sqrt(double)` המקבלת מספר ממשי ומחזירה את השורש הריבועי החיובי שלו. טיפוס ההחזרה של `Math.sqrt` הינו `double`. לדוגמא, הפונקציה `isCouple(3,4)` תחזיר את המספר 5.

```
public static int isCouple (int a, int b)
{
    int ans = -1;
    // השלימו בדף התשובות, שאלה 5 א'
    return ans;
}
```

סעיף ב' (8 נקודות)

השלימו את הפונקציה הסטטית הבאה המקבלת כקלט מספר שלם n , ומחזירה את מספר הזוגות הפיתגוראיים השונים כך ש $1 \leq a \leq n$ וגם $1 \leq b \leq n$. זוגות "שונים" הם זוגות שאינם "זהים". שימו לב: מכל שני זוגות זהים יש לקחת בחשבון רק אחד! לדוגמא, הפונקציה `howMany(4)` תחזיר את המספר 1, שכן קיים זוג פיתגוראי אחד בלבד, $\langle 3, 4 \rangle$ המקיים $3 \leq n$ וגם $4 \leq n$.

```
public static int howMany (int n)
{
    int ans = 0;
    // השלימו בדף התשובות, שאלה 5 ב'
    return ans;
}
```

המשך השאלה בדף הבא

סעיף ג' (10 נקודות)

השלימו את הפונקציה הסטטית הבאה המקבלת כקלט מספר שלם n ומחזירה את צפיפות (density) הזוגות הפיתגוראיים בטווח $1..n$, כאשר צפיפות זו מוגדרת כיחס בין מספר הזוגות הפיתגוראיים השונים שעבורם מתקיים $1 \leq a \leq n$ וגם $1 \leq b \leq n$, לבין מספר כל הזוגות השונים $\langle a, b \rangle$ הקיימים בטווח $1..n$ (יש לקחת בחשבון גם את הזוגות בהם $a=b$).

שימו לב: מכל שני זוגות מספרים זהים יש לקחת בחשבון רק אחד!

לדוגמא, הפונקציה `density(4)` תחזיר את הערך `0.1` משום שרק אחד מבין 10 הזוגות האפשריים בטווח (הזוגות `(1-1, 1-2, 1-3, 1-4, 2-2, 2-3, 2-4, 3-3, 3-4, 4-4)` הינו זוג פיתגוראי.

```
public static double density (int n)
{
    double ans = -1.0;
    // השלימו בדף התשובות, שאלה 5 ג'
    return ans;
}
```

שאלה שישיה (10 נקודות)

נתונה הפונקציה הבאה:

```
1 public static int [][] strategy (int[][] board, int player)
2 {
3     int [][] moves = possibleMoves(board, player);
4     if (moves.length > 0)
5     {
6         int [][] x = new int [moves.length][2];
7         int y=0, ansI=0;
8         for (int i=0; i<x.length; i=i+1)
9         {
10            for (int j=0; j<2; j=j+1)
11                x[i][j] = (int)Math.abs(moves[i][j]-(double)(size-1)/2);
12            if ((x[i][0]+x[i][1]+1) > y)
13            {
14                y=x[i][0]+x[i][1]+1;
15                ansI = i;
16            }
17        }
18        board = play(board, player, moves[ansI][0], moves[ansI][1]);
19    }
20    return board;
21 }
```

המשך השאלה בדף הבא

תזכורת מעבודת הבית השניה:

- size הינו משתנה סטטי (גלובלי), כך שמימדי הלוח board הינם size*size. ידוע כי size זוגי וגדול מ-2.
- הפונקציה possibleMoves(board,player) מחזירה מערך דו מימדי המכיל את כל מיקומי המהלכים האפשריים עבור השחקן פלייר על הלוח board, כך שהתא [i-1][0] במערך זה הינו הקואורדינטה row במהלך i והתא [i-1][1] הינו הקואורדינטה column במהלך ה i. במידה ולא קיימים מהלכים חוקיים אפשריים, יוחזר מערך ריק.
- הפונקציה play(board,player,row,column) מחזירה לוח המייצג את מצב הלוח לאחר מהלך של השחקן player במשבצת [row][column]. במידה והמהלך הנ"ל אינו חוקי, יוחזר board ללא כל שינוי.
- Math.abs(value) הינה פונקציה המחזירה את הערך המוחלט של value:
Math.abs(-4) → 4
Math.abs(3.2) → 3.2

סמן את האפשרויות הנכונות (תתכנן יותר מתשובה נכונה אחת)

- 1) בפונקציה ממומשת אסטרטגיית משחק הנותנת העדפה למהלכים הקרובים לקצוות הלוח board.
- 2) בפונקציה ממומשת אסטרטגיית משחק הנותנת העדפה למהלכים הקרובים למרכז הלוח board.
- 3) בפונקציה ממומשת אסטרטגיית משחק הנותנת העדפה למהלכים בחלק העליון של הלוח.
- 4) בפונקציה ממומשת אסטרטגיית משחק הנותנת העדפה למהלכים בחלק התחתון של הלוח.
- 5) יתכן כי עבור אותם קלטים בדיוק ובשימוש באותה פונקציה possibleMoves, תבחר הפונקציה הנ"ל מהלכים שונים בהרצות שונות.
- 6) יתכן כי עבור מימושים שונים (אך נכונים) של הפונקציה possibleMoves, תבחר הפונקציה הנ"ל מהלכים שונים.
- 7) החלפת שורה 11 בשורה:
$$x[i][j] = (int)Math.abs(moves[i][j] - (size-1)/2);$$
לא תשנה את האסטרטגיה.

שאלה שביעית (10 נקודות)

נתונות הפונקציות הבאות:

```
public static int [] what(int [] a, int [] b)
{
    int [] ans = new int [a.length+b.length];
    for (int i=0; i < a.length; i=i+1)
        ans[i]=a[i];
    int index = a.length;
    for (int i=0; i < b.length; i=i+1)
    {
        myInsert(ans,b[i],index);
        index=index+1;
    }
    return ans;
}

public static void myInsert(int[] array, int value, int i)
{
    while (i > 0 && array[i-1] > value)
    {
        array[i] = array[i-1];
        i = i-1;
    }
    array[i] = value;
}
```

לא ניתן להניח דבר לגבי הקלט.

סמן את האפשרויות הנכונות (תתכנן יותר מתשובה נכונה אחת)

- (1) קיימים קלטים עבורם נקבל שגיאה בזמן ריצה.
- (2) בהרצת הקוד, נקבל בהכרח שגיאה בזמן ריצה.
- (3) קיימים קלטים עבורם לא תסתיים ריצת התכנית (לולאה אינסופית).
- (4) הפונקציה what מחזירה מערך ממוין המכיל את כל איברי המערכים a ו-b גם במידה והמערכים a ו-b אינם ממוינים.
- (5) הפונקציה what מחזירה מערך ממוין המכיל את כל איברי המערכים a ו-b, רק במידה וגם איברי a וגם איברי b ממוינים בסדר עולה.
- (6) הפונקציה what מחזירה מערך ממוין המכיל את כל איברי המערכים a ו-b, רק במידה ואיברי a ממוינים בסדר עולה.
- (7) הפונקציה what מחזירה מערך ממוין המכיל את כל איברי המערכים a ו-b, רק במידה ואיברי b ממוינים בסדר עולה.
- (8) הפונקציה what מחזירה מערך ממוין המכיל את כל איברי המערכים a ו-b, רק במידה וכל איברי a גדולים יותר מכל איברי b.
- (9) נניח שהמערך x1 מכיל את כל הערכים בין 11 ל 50 ממוינים בסדר עולה, המערך x2 מכיל את כל הערכים בין 41 ל 50 ממוינים בסדר עולה, המערך y1 מכיל את כל הערכים בין 10 ל 1 ממוינים בסדר יורד והמערך y2 מכיל את כל הערכים בין 40 ל 1 ממוינים בסדר יורד. זמן הריצה של what(x1,y1) גבוה מזמן הריצה של what(x2,y2).

תזכורת:

- `int length()`
שיטה במחלקה `String` המחזירה את אורך המחרוזת
- `int indexOf(char c)`
שיטה במחלקה `String`, המחזירה את האינדקס הראשון במחרוזת, בו מופיע התו `c`, או `-1` במידה והתו כלל לא מופיע במחרוזת.
- `char charAt(int index)`
שיטה במחלקה `String`, המחזירה את התו במיקום `index` במחרוזת
- `String substring(int beginIndex, int endIndex)`
שיטה במחלקה `String` המחזירה מחרוזת חדשה, שהיא תת-מחרוזת של עצם המפתח בין התווים `beginIndex` ו-`endIndex`, (כולל התו ה-`beginIndex` ולא כולל התו ה-`endIndex`)
- `double Math.pow(double a, double b)`
פונקציה המחזירה את הערך של `a` בחזקת `b`
- `double Math.sqrt(double a)`
פונקציה המחזירה את השורש הריבועי החיובי של `a`
- `double Math.abs(double a)`
פונקציה המחזירה את ערכו המוחלט של `a`
- `int Math.abs(int a)`
פונקציה המחזירה את ערכו המוחלט של `a`