

סמסטר א' תשס"ז
בוזון ראשון – 24.11.06

דר' ג'יהאד אל-סנע
מר איתי בר-יוסף
גב' דיקלה דותן
פרופ' מיכאל קודיש
דר' יצחק רוזן
מר גיא שני

משך הבוחן שעתיים וחצי (לא תינתן הארכה)

חומר עזר - אסור

במבחן זה 6 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 ומסוגנת 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. במידה והוספת שורות הקוד של הסעיף תגרום לשגיאת קומפילציה או לשגיאה בזמן ריצה, יש לציין זאת במקום הקלט שיודפס למסך.
- **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 **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 נבחנו.

בהצלחה!

שאלה 1 (10 נקודות)

השלימו את הפונקציה `maxVal (int [][] arr)` המקבלת מערך דו מימדי `arr` של מספרים שלמים $0 \leq$, ומחזירה את ערך המספר המקסימלי (הגדול ביותר) במערך. תוכלו להניח שכל המערכים אינם `null` ושלפחות מערך אחד איננו ריק.

דוגמא:

בקבלת המערך `{ {1,8}, {}, {0,0,300,7,6}, {2,4,55,11} }`, יוחזר הערך 300.

```
public static int maxVal (int [][] arr) {
    int ans = 0;
    // השלימו בדף התשובות שאלה 1
    return ans;
}
```

שאלה 2 (16 נקודות: 4 נק' סעיף א' + 12 נק' סעיף ב')

השלימו את הפונקציה `countSameChar (String s1, String s2)` המקבלת שתי מחרוזות ומחזירה את מספר התווים הזהים המופיעים במיקום זהה בשתי המחרוזות. לא ניתן להניח דבר לגבי הקלט.

דוגמא:

בהנתן המחרוזות `abbcdbbtceaaa` יוחזר הערך 2.

בהנתן המחרוזות `abbcdbbceaaa` יוחזר הערך 3.

בהנתן המחרוזות `ababababab` יוחזר הערך 0.

```
public static int countSameChar (String s1, String s2) {
    // בדיקת חוקיות הקלט ( השלימו בדף התשובות שאלה 2, סעיף א )
    return 0;
    else {
        int count = 0;
        // השלימו בדף התשובות שאלה 2, סעיף ב
        return count;
    }
}
```

```

public static int[][] mysteriousPlayer(int[][] board, int player) {
    if ( !hasValidMoves(board, player) | canJump(board,player))
        board = randomPlayer(board, player);
    else {
        int [][] allMoves = getSimpleMoves(board, player);
        int [] bestMove = allMoves[0];
        int bestGrade = (int)Math.abs(allMoves[0][3]-3.5);
        for (int i=1; i<allMoves.length; i=i+1) {
            int thisGrade = (int)Math.abs(allMoves[i][3]-3.5);
            if (thisGrade>bestGrade) {
                bestMove = allMoves[i];
                bestGrade=thisGrade;
            }//end of if
        }//end of for
        board = applyMove(board, player, bestMove[0], bestMove[1],
            bestMove[2], bestMove[3]);
    }//end of else
    return board;
}

```

תזכורת מעבודת הבית השניה:

- הפונקציה `boolean hasValidMoves(int[][] board, int player)` , המחזירה ערך אמת באם השחקן יכול לנוע על הלוח (עם או ללא קפיצה), וערך שקר אחרת.
- הפונקציה `boolean canJump (int [][] board, int player)` מחזירה ערך אמת באם השחקן `player` יכול לבצע מהלך קפיצה על הלוח, וערך שקר אחרת.
- הפונקציה `int [][] randomPlayer(int [][] board, int player)` המקבלת לוח ושחקן ומחזירה את הלוח לאחר מהלך אקראי אך חוקי של השחקן. במידה ולא קיים כל מהלך אפשרי עבור `player`, לא ישונה הלוח.
- הפונקציה `int [][] getSimpleMoves(int [][] board, int player)` מחזירה מערך ובו כל ה"פשוטים" (הצעדים שאינם צעדי קפיצה) האפשריים. אין חשיבות לסדר הצעדים.
 במידה ויש `n` צעדים כאלה אפשריים, יוחזר מערך באורך `n`. המימד השני, יהיה 4: תאים 0,1 עבור קואורדינטות המקור (0 עבור שורה, 1 עבור עמודה) ותאים 2,3 עבור קואורדינטות היעד (2 עבור שורה, 3 עבור עמודה). במידה ולא קיימים מהלכים, יוחזר מערך ריק.

▪ הפונקציה `applyMove (int [][] board, int player, int fromRow, int fromCol, int toRow, int toCol)` המקבלת את הלוח, השחקן, קואורדינטות המקור וקואורדינטות היעד. הפונקציה משנה ומחזירה את לוח המשחק `board` בהתאם למהלך המבוצע. ניתן להניח כי המהלך חוקי.

☒ `Math.abs(value)` היא פונקציה המחזירה את הערך המוחלט של `value`:

`Math.abs(-4) → 4`

`Math.abs(3.2) → 3.2`

קירות הלוח, מסומנים בציור הבא בקו כפול ומודגש:

| | | | | | | | | |
|---|----|----|----|----|----|----|----|----|
| 7 | | -1 | | -1 | | -1 | | -1 |
| 6 | -1 | | -1 | | -1 | | -1 | |
| 5 | | -1 | | -1 | | -1 | | -1 |
| 4 | | | | | | | | |
| 3 | | | | | | | | |
| 2 | 1 | | 1 | | 1 | | 1 | |
| 1 | | 1 | | 1 | | 1 | | 1 |
| 0 | 1 | | 1 | | 1 | | 1 | |
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

עמודה

סמן את התשובות הנכונות (תיתכן יותר מתשובה נכונה אחת):

- בפונקציה ממומשת אסטרטגיית משחק הנותנת העדפה למהלכים המסתיימים בקרבת הקיר ה"שמאלי" באיור (במידה ולא ניתן לבצע "קפיצה")
- בפונקציה ממומשת אסטרטגיית משחק הנותנת העדפה למהלכים המסתיימים בקרבת אחד מקירות הלוח (במידה ולא ניתן לבצע "קפיצה")
- במידה וניתן לבצע מהלכי קפיצה, יבחר המהלך באופן אקראי
- בפונקציה ממומשת אסטרטגיית משחק הנותנת העדפה למהלכים בהם עמדת המקור הינה בסמוך לקיר ה"ימני" (במידה ולא ניתן לבצע "קפיצה")
- במידה וניתן לבצע מהלכי קפיצה, יבחר המהלך לפי אסטרטגיית משחק התקפית

שאלה 4 (12 נקודות):

נתונה הפונקציה הבאה:

```
public static void function(int[] array) {
    int from = 0;
    int to = array.length-1;
    while (from <= to) {
        int [] minMaxInd = findMinMaxIndex(array,from,to);
        swap(array,from,minMaxInd[0]);
        swap(array,to,minMaxInd[1]);
        from = from+1;
        to = to-1;
    } // end while
}
```

בנוסף, נתונה הפונקציה swap שנלמדה בכתה:

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

הקוד של הפונקציה `findMinMaxIndex (int [] arr, int a, int b)` אינו נתון, אך ידוע שפונקציה זו מקבלת שלושה פרמטרים, מערך של איברים מטיפוס `int` ושני אינדקסים לאיברים במערך (גם הם מטיפוס `int`). הפונקציה מחזירה מערך `int []` בעל שני תאים: בתא הראשון האינדקס לאיבר המינימלי בתת המערך `arr[a..b]`, ובתא השני האינדקס לאיבר המקסימלי בתת המערך `arr[a..b]`.

שים לב:

- תת המערך `arr[a..b]` הינו רצף התאים במערך `arr`, המתחיל באינדקס `a` ומסתיים באינדקס `b` (כולל `a` ו `b`)
- ניתן לסמוך על כך שהפונקציה הנ"ל נכונה.

סמן את התשובות הנכונות (תיתכן יותר מתשובה נכונה אחת):

- א. הפונקציה function תמיינ את איברי המערך array, רק במידה והמחצית הראשונה של המערך כבר ממוינת.
- ב. הפונקציה function תמיינ את איברי המערך array, רק במידה והמחצית השניה של המערך כבר ממוינת.
- ג. הפונקציה function תמיינ את איברי המערך array בסדר יורד.
- ד. הפונקציה function תמיינ את איברי המערך array בסדר עולה.
- ה. קיים קלט שעבורו תתרחש שגיאת ריצה במהלך ריצת הפונקציה function.
- ו. ניתן לשנות את התנאי בלולאת ה while ל (from < to) מבלי לשנות את תוצאת הפונקציה function.
- ז. ניתן לממש את הפונקציה findMinMaxIndex כך שתעבור על איברי המערך פעם אחת ויחידה.

הערה: הפונקציה function ממוינת נכונה (ובסדר עולה) רק חלק מהמערכים הניתנים לה. לדוגמא, קריאה לפונקציה עם המערך {8,5} תסתיים ללא שינוי המערך (זאת משום שאיברי המערך יוחלפו פעמיים: לאחר ההחלפה הראשונה יתקבל המערך {5,8}, אך ההחלפה השניה תחזיר את האיברים המקוריים למקומם). עם זאת, הוחלט בשל רמת השאלה הגבוהה כי סימון סעיף ד' כנכון, לא יגרור הפחתת נקודות.

שאלה 5 (20 נקודות)

סעיף א' (10 נקודות)

השלימו את הפונקציה הבוליאנית isPower(int a, int b) המקבלת שני מספרים שלמים a, b ומחזירה ערך אמת אם ורק אם b הינו חזקה (שלמה) של a. תוכלו להניח כי $a, b > 0$. לדוגמא:

isPower(2,8) \rightarrow true, (משום ש: $2^3 = 8$)

isPower(8,2) \rightarrow false, ($\forall n 8^n \neq 2$ משום ש:)

isPower(4,1) \rightarrow true, ($4^0 = 1$ משום ש:)

isPower(2,6) \rightarrow false, ($\forall n 2^n \neq 6$ משום ש:)

```
public static boolean isPower(int a, int b) {
    boolean ans;
    // השלימו בדף התשובות שאלה 5, סעיף א
    return ans;
}
```

סעיף ב' (10 נקודות)

השלימו את הפונקציה `powerOf(int b)` המקבלת מספר שלם `b` ומדפיסה למסך את כל המספרים השלמים והחיוביים ש `b` הינו חזקה שלהם. תוכלו להניח כי $b > 1$.

סדר ההדפסות אינו משנה.

תוכלו להשתמש בסעיף א' גם אם לא עניתם עליו.

לדוגמא:

```
powerOf(16) → 2
              4
              16
powerOf(13) → 13
```

```
public static void powerOf(int b) {
    // השלימו בדף התשובות שאלה 5, סעיף ב
}
```

שאלה 6 (30 נקודות):

נתונות הפונקציות הבאות:

```
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 = 2;
    int y = -2;
    int [] arr1 = {-4,-3,-2,-1,0};
    int [][] arr2 = {{2,4,6}, {4,5,6}, {2,4,6}, {1,2,3}};
    // פה יופיע קטע הקוד המופיע בכל סעיף
}
```

עבור כל אחד מהסעיפים הבאים סמנו בדף התשובות האם הוספת הקוד המופיע בסעיף במקום ההערה בקוד תגרוור:

א. שגיאת קומפילציה

ב. שגיאה בזמן ריצה

ג. פלט יודפס למסך. במקרה זה ציינו מה הפלט.

ד. אף אחד מהנ"ל.

משקל כל סעיף הינו 3 נקודות.

סעיף א'

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

סעיף ב'

```
System.out.println(arr2[0].length==arr2[1].length);
```

סעיף ג'

```
System.out.println(arr2[0]==arr2[2]);
```

סעיף ד'

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

סעיף ה'

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

סעיף ו'

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

סעיף ז'

```
int [] arr3 = new int [arr1[2]];  
System.out.println(arr3[0]);
```

סעיף ח'

```
System.out.println((int)arr2[2] / (double)arr1[0]);
```

סעיף ט'

```
double [] d1 = {1.0, 2.3, 7, 9};
```

סעיף י'

```
double [] d2 = {1.0, 3.4, 5.5};  
changeArr2(d2);  
System.out.println(d2[0]);
```


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`