

סמסטר א' תשע"א
בוחן אמצע – 3.12.10

פרופ' משה זיפר
מר שי זקוב
ד"ר פז כרמי
פרופ' מייק קודיש
ד"ר חן קיסר
ד"ר צחי רוזן

משך הבוחן: שעתיים
חומר עזר - אסור

במבחן זה 4 שאלות. ענו על כולן. בשאלות שבהן לא מצוין אחרת, ניתן לבחור בפתרון רקורסיבי או בפתרון שאינו רקורסיבי לבחירתכם.

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 ה חומר יועבר לגריסה.

שימו לב: החשוב ביותר הוא נכונות הקוד. עם זאת, יעילות, סגנון וכתובה ברורה חשובים גם הם ולכן תשובה יעילה ומסוגגנת תזכה בציון גבוה יותר

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**. כיתבו פתרון לשאלה קודם כול במחברת הטיטה, ורק לאחר מכן העתיקו פתרון נקי לדף התשובות

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

הקפידו לרשום בשני דפי התשובות גם את מספר הנבחן ואת מספר החדר שבו אתם נבחנים

אם אינכם יודעים את התשובה ל שאלה כלשהי, רשמו "לא יודע/ת" (במקום תשובה) ותזכו ב- 20% מניקוד השאלה.

שאלה 1 (25 נקודות)

השלימו את הפונקציה `diffValues(int[] a, int n)` אשר מקבלת מערך של מספרים שלמים וערך `n` גדול ממש-0 ומחזירה את הערך `true` אם ורק אם מתקיימות שתי התכונות הבאות:
1. כל איברי המערך שונים זה מזה

$$\forall i, j: (0 \leq i, j < a.length) \wedge (i \neq j) \Rightarrow (a[i] \neq a[j])$$

2. כל איברי המערך הינם בין הערך (כולל) לבין הערך `n` (כולל).

$$\forall i: (0 \leq i < a.length) \Rightarrow (1 \leq a[i] \leq n)$$

ניתן להניח כי מערך הקלט שונה מ-`null` ו-`n > 0`.

```
public static boolean diffValues(int[] a, int n) {
    boolean res;
    // Your code here.
    return res;
}
```

דוגמאות:

אם קלט הפונקציה הוא `a={5, 9, 2, 4, 8}` ו-`n=9` אזי יוחזר הערך `true`.
אם קלט הפונקציה הוא `a={1, 4, 1, 2}` ו-`n=4` אזי יוחזר הערך `false`.
אם קלט הפונקציה הוא `a={1, 2, 3, 9}` ו-`n=4` אזי יוחזר הערך `false`.

שאלה 2 (25 נקודות)

השלימו את הפונקציה `primeFactors(int n)` אשר מקבלת מספר שלם גדול ממש-1, ומדפיסה למסך את רשימת הגורמים הראשוניים שלו. יש להדפיס כל גורם פעם אחת בלבד, כאשר על הגורמים להיות מודפסים בסדר עולה.
ניתן להניח כי `n > 1`.

```
public static void primeFactors(int n) {
    // Your code here.
}
```

דוגמאות:

אם קלט הפונקציה הוא 396 (כאשר $396 = 2 * 2 * 3 * 3 * 11$), אזי תודפס השורה (שבה המספרים מופרדים על-ידי רווח יחיד):

2 3 11

אם קלט הפונקציה הוא 343 (כאשר $343 = 7 * 7 * 7$), אזי תודפס השורה:

7

שאלה 3 (25 נקודות)

במשחק "גפרורים 1,2,3" מונחת על השולחן ערימת גפרורים. שני שחקנים נוטלים מהערימה, לסירוגין, 1, 2 או 3 גפרורים. השחקן שנוטל את הגפרור האחרון הוא המנצח

חישבו על משחק, אני מולכם, ונניח שכעת תורי לשחק. אם יש על השולחן 1, 2, או 3 גפרורים - אני לוקח את כולם ומנצח. אם יש על השולחן 4 גפרורים, כל מהלך שלי יוביל למצב שבו אתם תוכלו לנצח. אם, לעומת זאת, יש 5 גפרורים, אוכל ליטול 1 מהם. כך יישארו על השולחן 4 גפרורים בלבד, וכל מהלך שלכם יוביל לניצחון שלי

במשחק הגפרורים, נאמר שיש לשחקן מהלך מנצח אם ורק אם הוא יכול ליטול מספר גפרורים כזה (על פי כללי המשחק), שמוותר אחת משתי האפשרויות: או שהוא מנצח מיד (לוקח את כל הגפרורים) או שנותר על השולחן מספר גפרורים כזה שליריבו לאהיה מהלך מנצח.

כיתבו פונקציה בוליאנית רקורסיבית `static boolean matches(int n)` בעלת התבנית הבאה, אשר מציינת האם יש לשחקן בתור הנוכחי מהלך מנצח במשחק "גפרורים 1,2,3" כאשר על השולחן n גפרורים.

```
public static boolean matches(int n) {
    boolean ans;
    // Your code here.
    return ans;
}
```

דוגמאות:

`matches(20)` תחזיר `false`
`matches(17)` תחזיר `true`

יש לכתוב פונקציה רקורסיבית יחידה (שלא קוראת לפונקציה אחרת), אשר מחשבת האם לשחקן שעכשיו תורו יש מהלך מנצח
אין להוסיף פונקציות נוספות. אין להוסיף משתנים נוספים. ניתן להניח ש $n > 0$.

שאלה 4 (25 נקודות)

נסמן בסימן \odot פעולה חשבונית על המספרים השלמים מבין הפעולות: חיבור (+) וחסור (-). בהינתן מערך (לא ריק) $a = \{a_0, a_1, \dots, a_{n-1}\}$ וערך `int value` יש להחליט האם קיים ביטוי חשבוני מהצורה $a_0 \odot a_1 \odot a_2 \odot \dots \odot a_{n-1}$ שערכו המחושב הינו `value`.

דוגמאות:

אם $a = \{2, 3, 6, 10\}$ ו `value = -5`, קיים ביטוי חשבוני כנדרש והוא $2 - 3 + 6 - 10$.
אם $a = \{5, 4, 6, 3\}$ ו `value = 11`, לא קיים ביטוי חשבוני כנדרש

כיתבו את הפונקציה הרקורסיבית `rec` בעלת התבנית הבאה:

```
public static boolean rec(int[] a, int value) {
    return rec(a.length-1, a, value);
}

public static boolean rec(int i, int[] a, int value) {
    boolean answer;
    // Your code here
    return answer;
}
```

ניתן להניח כי המערך `a` אינו `null` ומכיל לפחות איבר אחד.

תזכורת:

- `int length()`
שיטה במחלקה `String` המחזירה את אורך המחרוזת שמפעילה את השיטה.
- `int indexOf(char c)`
שיטה במחלקה `String` המחזירה את האינדקס הראשון במחרוזת שבו מופיע התו `c`, או -1 אם `c` אינו מופיע כלל במחרוזת
- `char charAt(int index)`
שיטה במחלקה `String` המחזירה את התו במיקום `index` במחרוזת שמפעילה את השיטה
- `String substring(int beginIndex)`
שיטה במחלקה `String`, המחזירה מחרוזת חדשה שהיא תת מחרוזת של המחרוזת שמפעילה את השיטה, החל מהמיקום ה-`beginIndex` (כולל) ועד המיקום האחרון (כולל).
- `String substring(int beginIndex, int endIndex)`
שיטה במחלקה `String`, המחזירה מחרוזת חדשה שהיא תת מחרוזת של המחרוזת שמפעילה את השיטה, החל מהמיקום ה-`beginIndex` (כולל) ועד המיקום ה-`endIndex` (לא כולל).
- `double Math.pow(double a, double b)`
פונקציה המחזירה את הערך `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`
- `boolean booleanValue()`
שיטה במחלקה `Boolean`, המחזירה ערך בוליאני המצין את הערך של אובייקט מסוג `Boolean`