

מבחן השלמה למגמת הנדסת תוכנה על-תיכונית

הוראות לנבחנים

א. משך הבחינה: שלוש שעות.

ב. מבנה השאלון ומפתח ההערכה: בשאלון זה שני פרקים ובהם חמש שאלות. עליכם לענות על כל השאלות. לכל שאלה – 20 נקודות. סך-הכול – 100 נקודות.

ג. חומר עזר מותר לשימוש: מחשבון פשוט שאינו ניתן לתכנות. כל חומר עזר כתוב בכתב-יד או מודפס על נייר.

ד. הוראות מיוחדות:

1. כתבו בעט בלבד.
2. ענו על השאלות במחברת הבחינה.
3. הדביקו את מדבקות הנבחן במקומות המיועדים לכך.

כתבו במחברת הבחינה בלבד, בעמודים נפרדים, כל מה שברצונכם לכתוב כטייטה (ראשי פרקים, חישובים וכדומה).
כתבו "טייטה" בראש כל עמוד טייטה. כתיבת טייטות כלשהן על דפים שמחוץ למחברת הבחינה עלולה לגרום לפסילת הבחינה!

בשאלון זה 8 עמודים.

השאלות בשאלון זה מנוסחות בלשון רבים, אף על פי כן על כל תלמידה וכל תלמיד להשיב עליהן באופן אישי.

בהצלחה!

השאלות

בשאלון זה שני פרקים ובהם חמש שאלות (לכל שאלה – 20 נקודות).
עליכם לענות על כל השאלות.

פרק ראשון (60 נקודות)

ענו על השאלות 1–3.

שאלה 1 (20 נקודות)

"ספרה המופיעה ברצף במספר" היא ספרה המופיעה בתוך מספר **יותר מפעם אחת**, ובסמיכות.

לדוגמה: הספרה 8 מופיעה ברצף במספר 7688854

הספרה 4 אינה מופיעה ברצף במספר 3472

(7 נק')

א.

לפניכם הפונקצייה SequenceInNum הכתובה בפסאודו־קוד.

הפונקצייה מקבלת ספרה חוקית (בין 0 ל-9) ומספר שלם חיובי, ומחזירה "אמת" אם הספרה שהתקבלה מופיעה במספר ב"רצף". אחרת, מחזירה "שקר".

בפונקצייה שלפניכם חסרים **חמישה** ביטויים, המסומנים בין סוגריים עגולים.

כתבו במחברת הבחינה את הביטויים החסרים (1)–(5), בסדר עולה, וכתבו ליד כל מספר את הביטוי החסר שהוא מייצג.

Boolean SequenceInNum (digit, num)

1. Boolean result ← False

2. while _____ (1)

2.1 if _____ (2)

then

2.1.1 _____ (3)

2.2 num ← _____ (4)

3. return _____ (5)

ב. (7 נק')

לפניכם הפונקצייה `DigitsNotInSequence` אשר מקבלת שני מספרים שלמים – `num1` ו-`num2`. הפונקצייה מחזירה מספר שלם חדש, המורכב מהספרות המופיעות ב-`num2` בסדר הפוך לסדר הופעתן בו, אך ללא הספרות המופיעות ברצף ב-`num1`. בפונקצייה שלפניכם חסרים **שישה** ביטויים, המסומנים בין סוגריים עגולים. כתבו במחברת הבחינה את הביטויים החסרים (6) – (11), בסדר עולה, וכתבו ליד כל מספר את הביטוי החסר שהוא מייצג.

```
Int DigitsNotInSequence(int num1,int num2)

1. int result ← _____ (6)

2. while _____ (7)
    2.1 if _____ (8)
        then
            2.1.1 result = _____ (9)
    2.2 num2 ← _____ (10)

3. return _____ (11)
```

ג. (6 נק')

סטודנט הציע לבעיה המובאת בסעיף א' פתרון אחר, המשתמש בפונקצייה `SequenceInNum2`. הפונקצייה `SequenceInNum2` מקבלת ספרה חוקית (בין 0 ל-9) ומספר שלם, ומחזירה את מיקום הספרה הראשונה ברצף הראשון. אם הספרה אינה מופיעה ברצף, יוחזר המיקום 0.

לדוגמה: עבור הזימון 8, 7688854 יוחזר הערך 3
עבור הזימון 4, 3472 יוחזר הערך 0

במידה שנשתמש ב-`SequenceInNum2` עבור כל שורה בפונקצייה `DigitsNotInSequence`, המובאת לעיל בסעיף ב', ענו במחברת הבחינה:

- האם השורה נשארת כמות שהיא, האם היא מבוטלת, או האם נדרש לשנות את השורה?
- בכל מקרה שעניתם כי נדרש לשנות את השורה – כתבו אותה מחדש באופן הנדרש.

שאלה 2 (20 נקודות)

במשרד התחבורה מחשבו מחדש את מערכת נתוני כלי הרכב.
לצורך כך הגדירו מחלקה כללית של כלי רכב בשם Car הכוללת את התכונות הבאות:

- מספר רישוי
- דגם
- שנת ייצור
- תקין / לא תקין
- אוסף הרכבים שבהם פגע הרכב כתוצאה מתאונות.

מלבד זאת, לכלי רכב מסוג משאית הוגדרה מחלקה מיוחדת בשם Truck הכוללת עוד שתי תכונות ופונקצייה:

- משקל ארגז המטען של המשאית
- הרכב שהיא גוררת (אם אינה גוררת רכב – הערך יהיה null)
- פונקציית carry, המפורטת להלן בסעיף ג'.

(6 נק') א. הגדירו את מחלקת משאית (Truck) תוך שימוש בהורשה, כולל תכונות, תיעוד ופעולה בונה שמאתחלת משאית שלא גוררת רכב. ציירו תרשים מחלקות UML מלא.

(5 נק') ב. במחלקת כלי רכב (Car) קיימת פעולה בשם BoomCarG, המוסיפה לרכב הנוכחי תאונה שבה פגע ברכב. הוסיפו למחלקה פעולה המחזירה את החלק היחסי של כמות המשאיות שפגעו ברכב, לעומת סך כל הרכבים שפגעו ברכב.

(4 נק') ג. למחלקת משאית (Truck) נוספה הפעולה שלהלן:

```
Public void carry (Car c)
{
}
```

הפעולה מעדכנת כי המשאית גוררת את הרכב, לפי התנאים הבאים:

- אם הרכב שהתקבל הוא משאית, ניתן לגרור אותו רק אם משקל הארגז של המשאית הנגררת קטן מרבע המשקל של ארגז המשאית הגוררת.
 - אם הרכב שהתקבל אינו משאית, ניתן לגרור אותו רק אם שנת הייצור שלו היא 2016 ואילך.
- בהתאם לתנאים אלו לפניכם סדרת הוראות:

ידוע כי car1 הוא רכב משנת 2018, ו-truck1 ו-truck2 הן משאיות.

```
car1.carry (truck1);
truck1.carry (truck2);
truck1.carry (truck1);
truck2.carry (car1);
```

עבור כל הוראה חוקית – הסבירו האם לא תוכל להתבצע גרירה?
עבור כל הוראה לא חוקית – כתבו את הסיבה לאי-החוקיות.

(5 נק') ד. הגדירו פעולה חיצונית המקבלת מערך של רכבים. לכל משאית שאינה גוררת רכב **כרגע**, חפשו את הרכב הראשון שעונה על הקריטריונים הבאים:

- האם הרכב לא תקין
 - האם המשאית יכולה לגרור אותו
- לאחר שנמצא רכב מתאים, יש לבצע את גרירתו על ידי המשאית הפנויה.
אם לא נמצא רכב כזה, אין לבצע דבר.

שאלה 3 (20 נקודות)

מערך נקרא 'מערך מאוזן', אם קיים במערך מספר n שעבורו: $n > arr.length / 2$, וכן $n < 1$, ומכפלת האיברים הנמצאים בשני "ראשיו" (ראש אחד מתחילת המערך והראש השני מסוף המערך) שבגודל n שוות זו לזו ושוות למכפלת כל האיברים שביניהם.

לדוגמה: $n = 3$

1	6	2	2	3	2	1	3	2	2
---	---	---	---	---	---	---	---	---	---

מכפלת שני הראשים בגודל $n = 3$ (צבועים באפור) שווה ל-12 בשני הצדדים, וכן מכפלת האיברים שביניהם (בלבן).

לפניכם הפונקצייה `findBalancedN` המקבלת מערך של מספרים שלמים (הניחו כי אורך המערך לפחות 5), ומחזירה את ערך ה- n המקסימלי שעבורו המערך יהיה מאוזן. אם אין ערך כזה, יש להחזיר ערך 0.

בפונקצייה שלפניכם חסרים אחד-עשר ביטויים, המסומנים בין סוגריים עגולים.

כתבו במחברת הבחינה את הביטויים החסרים (1)–(11), בסדר עולה, וכתבו ליד כל מספר את הביטוי החסר שהוא מייצג.

```
public static int findBalancedN(int[] arr)
{
    int len=arr.length;
    int firstProducts = _____ (1)
    int lastProducts = _____ (2)
    int middleProducts = 1;
    int n;
    int place=0 ;
    for(n=2; _____ (3) ; n++)
    {
        _____ (4) ;
    }
    if ( _____ (5) && _____ (6) )
        place = 2;
    for(n=2; _____ (7) ; n++)
    {
        firstProducts *= _____ (8)
        lastProducts *= _____ (9)
        middleProducts /= _____ (10)
        if (firstProducts==lastProducts && lastProducts==middleProducts)
            place = n+1;
    }
    return _____ (11) ;
}
```

פרק שני (40 נקודות)

ענו על השאלות 4–5.

שאלה 4 (20 נקודות)

נתונה רשימה ליניארית, המייצגת מספר שלם באורך כלשהו. כל צומת מכיל ספרה אחת בלבד. ספרת האחדות נמצאת בצומת האחרון של הרשימה.

לפניכם דוגמה לרשימה כמתואר לעיל:

- הרשימה `null->8->5->2` מייצגת את המספר 258.
- הרשימה `null->5->4->9->0->2->1` מייצגת את המספר 120,945.

נתונות שתי פונקציות, `reverse` ו-`add`.

הפונקצייה `reverse` מקבלת רשימה של ספרות ומחזירה אותה בסדר הפוך.

הפונקצייה `add` מקבלת שתי רשימות של ספרות, שכל אחת מהן היא רשימה ליניארית כמתואר לעיל, ומחזירה רשימה המייצגת את החיבור בין שני מספרים נומריים שלמים חיוביים.

בשתי הפונקציות שלפניכם חסרים **אחד-עשר** ביטויים, המסומנים בין סוגריים עגולים. כתבו במחברת הבחינה את הביטויים החסרים (1)–(11), בסדר עולה, וכתבו ליד כל מספר את הביטוי החסר שהוא מייצג.

פונקצייה 1:

```
public static List<Integer> reverse(List<Integer> list) {  
    List<Integer> reversed = new ArrayList();  
    for (int i = list.size() - 1; i >= 0; i--) {  
        _____ (1)  
    }  
    return reversed;  
}
```

פונקציה 2:

```
public static List<Integer> add(List<Integer> list1, List<Integer> list2) {  
    List<Integer> result = new ArrayList<>();  
  
    list1 = _____ (2) _____;  
    list2 = _____ (3) _____;  
    Iterator<Integer> it1 = _____ (4) _____;  
    Iterator<Integer> it2 = _____ (5) _____;  
  
    int carry = 0;  
    while ( _____ (6) _____ ) {  
        int sum = carry;  
        if (it1.hasNext()) {  
            _____ (7) _____;  
        }  
        if (it2.hasNext()) {  
            _____ (8) _____;  
        }  
        _____ (9) _____  
        carry = _____ (10) _____;  
    }  
    return _____ (11) _____;  
}
```

שאלה 5 (20 נקודות)

(14 נק') א. לפניכם הפונקצייה `isReversed`, המקבלת מחרוזת בפורמט: `AcB`, (כאשר האותיות `A` ו-`B` מייצגות תתי מחרוזות ו-`c` הוא תו במחרוזת המופיע פעם אחת בלבד), ובודקת אם תתי-מחרוזות `A` היא הופכית לתתי-מחרוזת `B`.

בפונקצייה שלפניכם חסרים **שבעה** ביטויים, המסומנים בין סוגריים עגולים. כתבו במחברת הבחינה את הביטויים החסרים (1)–(7), בסדר עולה, וכתבו ליד כל מספר את הביטוי החסר שהוא מייצג.

```
public static boolean isReversed(String s) {  
    Stack<Character> st= new Stack<Character>();  
    Queue<Character> q= new LinkedList<Character>();  
    boolean flag = true;  
    boolean sep = false;  
    if _____ (1) _____  
        flag = false;  
    char ch;  
    for (int i = 0; _____ (2) _____; i++) {  
        ch = s.charAt(i);  
        if (s.charAt(i)=='c')  
            _____ (3) _____;  
        else  
            if (!sep)  
                _____ (4) _____;  
            else _____ (5) _____;  
    }  
    while (_____ (6) _____) {  
        if (_____ (7) _____)  
            flag = false;  
    }  
    return flag;  
}
```

(6 נק') ב. האם ניתן לפתור את התרגיל רק על ידי מחסניות, ללא שימוש בתור? נמקו את תשובתכם.

בהצלחה!