

## מדעי המחשב

### על פי תכנית הרפורמה ללמידה משמעותית

#### הוראות לנבחן

- א. משך הבחינה: שלוש שעות.
- ב. מבנה השאלון ומפתח ההערכה: בשאלון זה שלושה פרקים.  
פרק ראשון – בפרק זה שלוש שאלות, ענה על פי ההוראות בפרק.  
פרק שני – בפרק זה שלוש שאלות, ומהן עליך לענות על שתיים.  
פרק שלישי – בפרק זה שאלות בארבעה מסלולים שונים.  
ענה על שאלה אחת במסלול שלמדת: 25 נקודות  
סה"כ – 100 נקודות

ג. חומר עזר מותר בשימוש: כל חומר עזר, חוץ ממחשב הניתן לתכנות.

ד. הוראות מיוחדות:

- את כל התכניות שאתה נדרש לכתוב בשפת מחשב בפרקים הראשון והשני כתוב בשפה אחת בלבד – Java או C#.
- רשום על הכריכה החיצונית של המחברת באיזו שפה אתה כותב – Java או C#.
- רשום על הכריכה החיצונית של המחברת את שם המסלול שלמדת.  
המסלול הוא אחד מארבעת המסלולים האלה:  
מערכות מחשב ואסמבלי, מבוא לחקר ביצועים, מודלים חישוביים, תכנות מונחה עצמים.  
הערה: בתכניות שאתה כותב לא יורדו לך נקודות, אם תכתוב אות גדולה במקום אות קטנה או להפך.

כתוב במחברת הבחינה בלבד, בעמודים נפרדים, כל מה שברצונך לכתוב בטייטה (ראשי פרקים, חישובים וכדומה).  
רשום "טייטה" בראש כל עמוד טייטה. רישום טייטות בלשהן על דפים שמחוץ למחברת הבחינה עלול לגרום לפסילת הבחינה!

ההנחיות בשאלון זה מנוסחות בלשון זכר ומכוונות לנבחנות ולנבחנים כאחד.

**בהצלחה!**

/המשך מעבר לדף/

## השאלות

בשאלון זה שלושה פרקים.

עליך לענות על שאלות משלושת הפרקים, לפי ההוראות בכל פרק.

### פרק ראשון (25 נקודות)

הערה: בכל שאלה שנדרשת בה קליטה, אין צורך לבדוק את תקינות הקלט.

לפותרים ב-Java: בכל שאלה שנדרשת בה קליטה, הנח שבתכנית כתובה ההוראה:

```
Scanner input = new Scanner (System.in);
```

ענה על שאלה 1 – חובה (10 נקודות)

1. נתונה פעולה ראשית במחלקה `Stam`. בפעולה הראשית הוגדר מערך חד-ממדי `arr` מטיפוס שלם.

כתוב ב-Java או ב-C#, בפעולה הראשית, קטע תכנית שיקלוט מספר שלם `num`, וידפיס את סכום כל האיברים במערך שערכם קטן מ-`num`.

/המשך בעמוד 3/

ענה על אחת מהשאלות 2 - 3 (15 נקודות)

2. הגדירו מחלקה A בעלת שתי תכונות: n1 – מטיפוס שלם ו־ n2 – מטיפוס שלם.

במחלקה הוגדרו: שתי פעולות בונות,

ולכל תכונה פעולות get ו־ set ב־ Java ו־ Get ו־ Set ב־ C# .

במחלקה זו הפעולה toString ב־ Java ו־ ToString ב־ C# מחזירה מחרוזת שהיא תרגיל החיבור

של ערכי n1 ו־ n2, והתוצאה שלו.

לדוגמה:

אם הערך שב־ n1 הוא המספר 3 והערך שב־ n2 הוא המספר 9, תחזיר הפעולה את המחרוזת:

"3 + 9 = 12"

כמו כן הפעולה void add(int k) ב־ Java ו־ void Add(int k) ב־ C# מוסיפה את הערך k

לערך של כל אחת מן התכונות n1 ו־ n2.

הדפסת עצם מזמנת ב־ Java את הפעולה toString של המחלקה וב־ C# את הפעולה ToString

של המחלקה.

לפניך הפעולה הראשית במחלקה Program כתובה ב־ Java וב־ C#.

הפעולה משתמשת במחלקה A.

### Java

```
public class Program
{
    public static void main(String[] args)
    {
        A a = new A();
        a.setN1(4);
        a.setN2(5);
        A a1 = new A(a.getN1(), a.getN2());
        System.out.println(a);
        a1.add(4);
        System.out.println(a);
        System.out.println(a1);
    }
}
```

### C#

```
public class Program
{
    public static void Main()
    {
        A a = new A();
        a.SetN1(4);
        a.SetN2(5);
        A a1 = new A(a.GetN1(), a.GetN2());
        Console.WriteLine(a);
        a1.Add(4);
        Console.WriteLine(a);
        Console.WriteLine(a1);
    }
}
```

(שים לב: המשך השאלה בעמוד הבא.)

א. עליך להגדיר את המחלקה A על פי התת-סעיפים v-i כך שהפעולה הראשית תתבצע בלי שגיאות הידור ו/או ריצה.

i כתוב ב-Java או ב-C# את כותרת המחלקה ואת התכונות שלה.

ii כתוב ב-Java או ב-C# את הכותרות של שתי הפעולות הבונות של המחלקה A.

iii ממש ב-Java את כל הפעולות get ו-set או ב-C# את כל הפעולות Get ו-Set הנחוצות לריצה של הפעולה הראשית.

iv ממש ב-Java את הפעולה add או ב-C# את הפעולה Add.

v ממש ב-Java את הפעולה toString או ב-C# את הפעולה ToString.

ב. הצג את העצמים הנוצרים עקב הפעלת הפעולה הראשית. לכל עצם רשום את השמות של כל התכונות שלו ואת הערכים של כל התכונות.

כמו כן הצג את השינויים בעצמים בעקבות ההוראה add(4) ב-Java או Add(4) ב-C#. רשום את הפלט של הפעולה הראשית.

/המשך בעמוד 5/

3. לפיך ממשק המחלקה **Button** המייצגת כפתור. לכפתור שלוש תכונות:  
מספר הרשום עליו – num, גודל הכפתור – size וצבע הכפתור – color.

כותרת הפעולה ב־ Java	תיאור הפעולה
public Button(int num)	פעולה הבונה כפתור שרשום עליו המספר num, בגודל 5, בצבע "black".
public Button(int num, int size, String color)	פעולה הבונה כפתור שרשום עליו המספר num, בגודל size, בצבע color.
public int getSize()	פעולה המחזירה את גודל הכפתור.
public void addToSize(int x)	פעולה המגדילה את גודל הכפתור ב־ x.
public boolean isSameSize(Button other)	פעולה המחזירה true אם הכפתור other זהה בגודלו לכפתור הנוכחי, אחרת – הפעולה מחזירה false.

כותרת הפעולה ב־ C#	תיאור הפעולה
public Button(int num)	פעולה הבונה כפתור שרשום עליו המספר num, בגודל 5, בצבע "black".
public Button(int num, int size, string color)	פעולה הבונה כפתור שרשום עליו המספר num, בגודל size, בצבע color.
public int GetSize()	פעולה המחזירה את גודל הכפתור.
public void AddToSize(int x)	פעולה המגדילה את גודל הכפתור ב־ x.
public bool IsSameSize(Button other)	פעולה המחזירה true אם הכפתור other זהה בגודלו לכפתור הנוכחי, אחרת – הפעולה מחזירה false.

א. ממש במחלקה **Button**, ב־ Java או ב־ C#, את הפעולה הבונה Button(int num).

ב. ממש במחלקה **Button**, ב־ Java או ב־ C#, את הפעולה המגדילה את גודל הכפתור ב־ x.

ג. ממש במחלקה **Button**, ב־ Java או ב־ C#, את הפעולה הבודקת אם גודל הכפתור

other זהה לגודל הכפתור הנוכחי.

(שים לב: סעיף ד של השאלה בעמוד הבא.)

7. לפיך קטע מפעולה ראשית במחלקה Program הכתוב ב־ Java וב־ C#.

עקוב אחר ביצוע קטע הפעולה, ורשום את הפלט שיתקבל. לכל עצם רשום את השמות של כל התכונות שלו, את הערכים של כל התכונות ואת השינויים שהיו בהם.

### Java

```
Button a1 = new Button(1, 12, "red");
Button a2 = new Button(2, 12, "green");
Button a3 = new Button(3, 12, "blue");
Button a4 = new Button(4);
a1.addToSize(2);
a4.addToSize(9);
if (a1.isSameSize(a3))
    System.out.println("***");
if (a4.isSameSize(a1))
    System.out.println("$$$");
if (a2.isSameSize(a3))
    System.out.println("###");
```

### C#

```
Button a1 = new Button(1, 12, "red");
Button a2 = new Button(2, 12, "green");
Button a3 = new Button(3, 12, "blue");
Button a4 = new Button(4);
a1.AddToSize(2);
a4.AddToSize(9);
if (a1.IsSameSize(a3))
    Console.WriteLine("***");
if (a4.IsSameSize(a1))
    Console.WriteLine("$$$");
if (a2.IsSameSize(a3))
    Console.WriteLine("###");
```

/המשך בעמוד 7/

**פרק שני (50 נקודות)**

**שים לב:** בכל שאלה שנדרש בה מימוש אתה יכול להשתמש בפעולות של המחלקות תור, מחסנית, עץ בינרי וחוליה, בלי לממש אותן. אם אתה משתמש בפעולות נוספות, עליך לממש אותן.

ענה על שתיים מהשאלות 4-6 (לכל שאלה — 25 נקודות).

4. רשימה L תיקרא **משולשת** אם היא מקיימת את התנאים האלה:

\* הרשימה אינה ריקה.

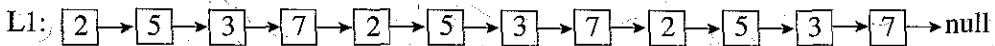
\* מספר האיברים בה מתחלק ב-3 בלי שארית.

\* האיברים בשליש הראשון של הרשימה מכילים את אותם ערכים שמכילים האיברים

בשליש השני של הרשימה ואותם ערכים שמכילים האיברים בשליש השלישי של הרשימה.

הערכים מסודרים באותו סדר בכל אחד מהשלישים.

לדוגמה: הרשימה L1 שלפניך היא **משולשת** באורך 12.



כתוב ב- C# או ב- Java פעולה חיצונית שתקבל רשימה L שהאיברים שלה הם מטיפוס שלם.

אם L היא רשימה **משולשת**, הפעולה תחזיר true.

אחרת — הפעולה תחזיר false.

/המשך בעמוד 8/

5. שים לב: לשאלה זו שני נוסחים: אחד ב-Java (עמודים 8-9), ואחד ב-C# (עמודים 10-11). עבוד על פי השפה שלמדת.

### לפתורים ב-Java

לפניך הפעולות sod ו-what המקבלות מערך a שאיבריו מטיפוס שלם, ממוין בסדר עולה, ומספר שלם k. לשתי הפעולות אותה טענת יציאה.

```
public static boolean sod(int[] a , int k)
{
    for (int i = 0; i < a.length-1; i++)
    {
        int j = i+1;
        while (j < a.length)
        {
            if (a[i] + a[j] == k)
                return true;
            j++;
        }
    }
    return false;
}
```

```
public static boolean what(int[] a , int k)
{
    int left = 0 , right = a.length-1;
    while (left < right)
    {
        if (a[left] + a[right] == k)
            return true;
        if (a[left] + a[right] < k)
            left++;
        else
            right--;
    }
    return false;
}
```



נתון מערך  $a$ : 

2	4	7	12	18
---	---	---	----	----

א. עקוב בעזרת טבלת מעקב אחר ביצוע הפעולה sod בעבור המערך הנתון  $a$ .  
המספר  $k = 11$ , רשום את הערך המוחזר.

בטבלת המעקב יש לכלול עמודות בעבור:  $a[i]$ ,  $a[j]$ ,  $i$ ,  $j$ , ועמודה נוספת שבה יצוין אם התנאי שבפקודת if מתקיים או אינו מתקיים.

ב. עקוב בעזרת טבלת מעקב אחר ביצוע הפעולה sod בעבור המערך הנתון  $a$ .  
המספר  $k = 10$ , רשום את הערך המוחזר.

בטבלת המעקב יש לכלול את העמודות שפורטו בסעיף א.

ג. מהי טענת היציאה של הפעולה sod?

ד. מהי סיבוכיות זמן הריצה של הפעולה sod? נמק את תשובתך.

ה. עקוב בעזרת טבלת מעקב אחר ביצוע הפעולה what בעבור המערך הנתון  $a$ .  
המספר  $k = 11$ , רשום את הערך המוחזר.

בטבלת המעקב יש לכלול עמודות בעבור:  $a[left]$ ,  $a[right]$ , left, right,

ושתי עמודות נוספות לכל אחת מפקודות if, בכל עמודה יצוין אם התנאי בפקודת if מתקיים או אינו מתקיים.

ו. מהי סיבוכיות זמן הריצה של הפעולה what? נמק את תשובתך.

ז. מי מבין שתי הפעולות sod או what יעילה יותר? נמק את תשובתך.

ח. טענת הכניסה של הפעולות sod ו-what שונתה כך שאפשר להעביר אליהן מערך  $a$  לא ממוין.

(1) האם טענת היציאה של הפעולה sod תשתנה? נמק את תשובתך.

(2) האם טענת היציאה של הפעולה what תשתנה? נמק את תשובתך.

לפותרים ב- C#

לפניך הפעולות Sod ו- What המקבלות מערך a שאיבריו מטיפוס שלם, ממוין בסדר עולה, ומספר שלם k. לשתי הפעולות אותה טענת יציאה.

```
public static bool Sod(int[] a , int k)
{
    for (int i = 0; i < a.Length-1; i++)
    {
        int j = i+1;
        while (j < a.Length)
        {
            if (a[i] + a[j] == k)
                return true;
            j++;
        }
    }
    return false;
}
```

```
public static bool What(int[] a , int k)
{
    int left = 0 , right = a.Length-1;
    while (left < right)
    {
        if (a[left] + a[right] == k)
            return true;
        if (a[left] + a[right] < k)
            left++;
        else
            right--;
    }
    return false;
}
```

נתון מערך  $a$ : 

2	4	7	12	18
---	---	---	----	----

א. עקוב בעזרת טבלת מעקב אחר ביצוע הפעולה Sod בעבור המערך הנתון  $a$ .

המספר  $k = 11$ . רשום את הערך המוחזר.

בטבלת המעקב יש לכלול עמודות בעבור:  $a[i]$ ,  $a[j]$ ,  $i$ ,  $j$ , ועמודה נוספת שבה יצוין אם

התנאי שבפקודת if מתקיים או אינו מתקיים.

ב. עקוב בעזרת טבלת מעקב אחר ביצוע הפעולה Sod בעבור המערך הנתון  $a$ .

המספר  $k = 10$ . רשום את הערך המוחזר.

בטבלת המעקב יש לכלול את העמודות שפורטו בסעיף א.

ג. מהי טענת היציאה של הפעולה Sod?

ד. מהי סיבוכיות זמן הריצה של הפעולה Sod? נמק את תשובתך.

ה. עקוב בעזרת טבלת מעקב אחר ביצוע הפעולה Sod בעבור המערך הנתון  $a$

והמספר  $k = 11$ . רשום את הערך המוחזר.

בטבלת המעקב יש לכלול עמודות בעבור:  $a[\text{left}]$ ,  $a[\text{right}]$ ,  $\text{left}$ ,  $\text{right}$ ,

ושתי עמודות נוספות לכל אחת מפקודות if. בכל עמודה יצוין אם התנאי

בפקודת if מתקיים או אינו מתקיים.

ו. מהי סיבוכיות זמן הריצה של הפעולה Sod? נמק את תשובתך.

ז. מי מבין שתי הפעולות Sod או What – יעילה יותר? נמק את תשובתך.

ח. טענת הכניסה של הפעולות Sod ו-What שונתה כך שאפשר להעביר אליהן

מערך  $a$  לא ממוין.

(1) האם טענת היציאה של הפעולה Sod תשתנה? נמק את תשובתך.

(2) האם טענת היציאה של הפעולה Sod תשתנה? נמק את תשובתך.

**שיום לב:** לשאלה זו שני נוסחים: אחד ב־ Java (עמודים 12-13), ואחד ב־ C# (עמודים 14-15). עבוד על פי השפה שלמדת.

### לפותרים ב־ Java

המשחק 'שחק נא' הוא משחק לשחקן אחד בשני שלבים. השחקן מקבל 52 קלפים מעורבבים, שונים זה מזה. על כל קלף יש מספר בין 1 ל־ 13, וצויר של אחת מארבע צורות. הצורות מיוצגות על ידי המספרים 1 עד 4. בשלב הראשון השחקן מחלק את הקלפים לארבע ערמות על פי הצורה של הקלף, כך שבכל ערמה יש קלפים עם אותה צורה. הקלפים מונחים זה על גבי זה. בשלב השני השחקן מגריל מספר בין 1 ל־ 4 המייצג צורה של קלף. השחקן פונה לערמה שעל כל הקלפים בה נמצאת הצורה שמיוצגת על ידי המספר שהוגרל. הוא מרים את הקלף שבראש הערמה, ומעביר אותו לערמה חמישית. שלב זה יתבצע עד שהמשחק יסתיים. המשחק מסתיים כאשר מוגרל מספר המייצג ערמה שאין בה קלפים. ניצחון הוא מצב שבו סכום המספרים שעל הקלפים בערמה החמישית מתחלק ב־ 100 ללא שארית.

לאחר שהמשחק מסתיים בודקים אם יש ניצחון.

לצורך מימוש המשחק הוגדרו המחלקות: **Card** המייצגת קלף אחד, **Deck** המייצגת את חמש הערמות הנדרשות במשחק, ו־ **Test** המנהלת את המשחק. כמו כן נכתבה במחלקה **Test** הפעולה `public static boolean game(Card[] cards)`, המקבלת מערך של 52 קלפים שונים המסודרים באופן אקראי. הפעולה מנהלת את המשחק עד סיומו. הפעולה מחזירה `true` אם המשחק הסתיים בניצחון, ואחרת – `false`. לפניך המחלקה **Card** המייצגת קלף שעליו מספר (value) בין 1 ל־ 13 וצורה (shape) בין 1 ל־ 4.

```
public class Card
{
    private int value;
    private int shape;

    public Card (int value, int shape)
    {
        this.value = value;
        this.shape = shape;
    }

    public int getValue () { return this.value; }
    public int getShape () { return this.shape; }
}
```

/המשך בעמוד 13/

א. i כתוב ב־ Java את כותרת המחלקה **Deck** ואת התכונות שלה.

ii כתוב ב־ Java במחלקה **Deck** את כותרות הפעולות האלה:

— פעולה בונה המגדירה את חמש הערמות להיות ריקות.

— פעולה insert המקבלת קלף ומוסיפה אותו לערמה הנכונה על פי השלב

הראשון במשחק.

— פעולה move המגרילה מספר המייצג צורה. אם ערמת הקלפים שצורתה

הוגרלה היא ריקה, הפעולה תחזיר false. אחרת — הפעולה תעביר לערמה

החמישית את הקלף שבראש הערמה שמספרה הוגרל ותחזיר true.

— פעולה sum המחזירה את הסכום הנוכחי של המספרים שעל הקלפים

בערמה החמישית.

שים לב: בתת־סעיף זה אין צורך לממש את הפעולות.

ב. ממש ב־ Java את הפעולה הבונה במחלקה **Deck**.

ג. ממש ב־ Java את הפעולה sum במחלקה **Deck**.

ד. ממש ב־ Java את הפעולה game במחלקה **Test**.

אתה יכול להשתמש בפעולות insert ו־ move של המחלקה **Deck**, בלי לממש אותן. אם

אתה משתמש בפעולות נוספות, עליך לממש אותן.

לפותרים ב- C#

המשחק 'שחק נא' הוא משחק לשחקן אחד בשני שלבים. השחקן מקבל 52 קלפים מעורבבים, שונים זה מזה. על כל קלף יש מספר בין 1 ל-13, וצויר של אחת מארבע צורות. הצורות מיוצגות על ידי המספרים 1 עד 4.

בשלב הראשון השחקן מחלק את הקלפים לארבע ערמות על פי הצורה של הקלף, כך שבכל ערמה יש קלפים עם אותה צורה. הקלפים מונחים זה על גבי זה.

בשלב השני השחקן מגריל מספר בין 1 ל-4 המייצג צורה של קלף. השחקן פונה לערמה שעל כל הקלפים בה נמצאת הצורה שמיוצגת על ידי המספר שהוגרל. הוא מרים את הקלף שבראש הערמה, ומעביר אותו לערמה חמישית.

שלב זה יתבצע עד שהמשחק יסתיים.

המשחק מסתיים כאשר מוגרל מספר המייצג ערמה שאין בה קלפים.

ניצחון הוא מצב שבו סכום המספרים שעל הקלפים בערמה החמישית מתחלק ב-100 ללא שארית.

לאחר שהמשחק מסתיים בודקים אם יש ניצחון.

לצורך מימוש המשחק הוגדרו המחלקות: **Card** המייצגת קלף אחד, **Deck** המייצגת את חמש הערמות הנדרשות במשחק, ו- **Test** המנהלת את המשחק.

כמו כן נכתבה במחלקה **Test** הפעולה `public static bool Game(Card[] cards)`, המקבלת מערך של 52 קלפים שונים המסודרים באופן אקראי. הפעולה מנהלת את המשחק עד סיומו. הפעולה מחזירה `true` אם המשחק הסתיים בניצחון, ואחרת – `false`.

לפניך המחלקה **Card** המייצגת קלף שעליו מספר (value) בין 1 ל-13 וצורה (shape) בין 1 ל-4.

```
public class Card
{
    private int value;
    private int shape;

    public Card (int value, int shape)
    {
        this.value = value;
        this.shape = shape;
    }

    public int GetValue () { return this.value; }
    public int GetShape () { return this.shape; }
}
```

i. כתוב ב-C# את כותרת המחלקה **Deck** ואת התכונות שלה.

ii. כתוב ב-C# במחלקה **Deck** את כותרות הפעולות האלה:

— פעולה בונה המגדירה את חמש הערמות להיות ריקות.

— פעולה **Insert** המקבלת קלף ומוסיפה אותו לערמה הנכונה על פי השלב

הראשון במשחק.

— פעולה **Move** המגרילה מספר המייצג צורה. אם ערמת הקלפים שצורתה

הוגרלה היא ריקה, הפעולה תחזיר **false**. אחרת — הפעולה תעביר לערמה

החמישית את הקלף שבראש הערמה שמספרה הוגרל ותחזיר **true**.

— פעולה **Sum** המחזירה את הסכום הנוכחי של המספרים שעל הקלפים

בערמה החמישית.

שים לב: בתת-סעיף זה אין צורך לממש את הפעולות.

ב. ממש ב-C# את הפעולה הבונה במחלקה **Deck**.

ג. ממש ב-C# את הפעולה **Sum** במחלקה **Deck**.

ד. ממש ב-C# את הפעולה **Game** במחלקה **Test**.

אתה יכול להשתמש בפעולות **Insert** ו-**Move** של המחלקה **Deck** בלי לממש אותן, אם

אתה משתמש בפעולות נוספות, עליך לממש אותן.

**פרק שלישי (25 נקודות)**

בפרק זה שאלות בארבעה מסלולים:  
 מערכות מחשב ואסמבלי, עמודים 16-17.  
 מבוא לחקר ביצועים, עמודים 18-24.  
 מודלים חישוביים, עמוד 25-26.  
 תכנות מונחה עצמים ב־ Java, עמודים 27-31; תכנות מונחה עצמים ב־ C#, עמודים 32-36.  
**ענה על שאלה אחת במסלול שלמדת.**

**מערכות מחשב ואסמבלי**

אם למדת מסלול זה, ענה על אחת מהשאלות 7-8 (25 נקודות).

7. בשאלה זו שני סעיפים, א-ב, שאין קשר ביניהם. ענה על שניהם.

א. לפניך קטע תכנית באסמבלי:

שים לב: הנתונים הם הקסדצימליים.

```
START: MOV     AX , C83BH
        MOV     BX , A89CH
        SHL    AX , 1
        OR     AL , 33H
        NOT    BL
        ADD    AX , BX
```

עקוב בעזרת טבלת מעקב אחר ביצוע קטע התכנית.

בטבלת המעקב פרט את התוכן של AX, BX, ZF, SF, CF בכל שלב.

/המשך בעמוד 17/



ב. (אין קשר לסעיף א.)

במקטע הנתונים הוגדר מערך:

ARR DB 5 DUP(?)

לפניך 4 קטעים iv-i באסמבלי שמטרתם לאתחל ל-0 את כל תאי המערך ARR. עקוב בעזרת טבלת מעקב אחרי הביצוע של כל אחד מהקטעים iv-i שלפניך, וקבע אם הוא מבצע את הנדרש או אינו מבצע את הנדרש.

<b>i</b>		MOV	SI, 0
		MOV	CX, 4
	A1:	MOV	ARR[SI], 0
		INC	SI
		LOOP	A1

---

<b>ii</b>		MOV	CX, 5
		LEA	BX, ARR
		MOV	AL, 0
	A1:	MOV	[BX], AL
		INC	BX
	LOOP	A1	

---

<b>iii</b>		MOV	BX, 5
		DEC	BX
	A1:	MOV	ARR[BX], 0
		DEC	BX
		JNZ	A1

---

<b>iv</b>		MOV	DI, 0
	A1:	MOV	ARR[DI], 0
		INC	DI
		CMP	DI, 5
		JC	A1

8. באוגר AX מאוחסן מספר בינרי. כתוב קטע תכנית באסמבלי, שיציב באוגר BL את מספר הפעמים שהרצף 1011 מופיע במספר הבינרי שבאוגר AX.

לדוגמה: אם באוגר AX מאוחסן המספר הבינרי: 0110110010110110

יוצב באוגר BL המספר 3.

**מבוא לחקר ביצועים**

אם למדת מסלול זה, ענה על אחת מהשאלות 9-10 (25 נקודות).

9. נתונה בעיית תכנון לינארי:

$$\text{Max } \{z = 5x_1 - x_2\}$$

בכפוף לאילוצים האלה:

$$- 2x_1 + 3x_2 \leq 18$$

$$x_1 - x_2 \leq 2$$

$$x_1 \leq 3$$

$$x_2 \geq 0$$

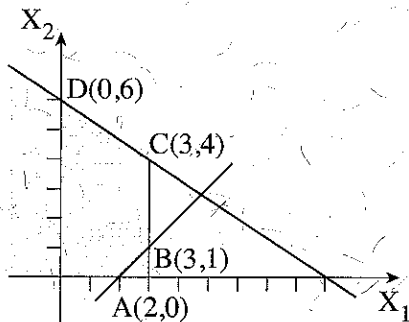
$x_1$  אינו מוגבל בסימן.

נתון גם כי הפתרון  $(3, 1)$  הוא פתרון אופטימלי של הבעיה.

כל אחד מהסעיפים א-ה שלפניך מתייחס לבעיית התכנון הלינארי הנתונה.

הסעיפים אינם תלויים זה בזה. ענה על כל הסעיפים.

א. לפניך סרטוט של תחום-הפתרונות האפשריים של הבעיה הנתונה.



העתק למחברתך את הסרטוט, ובצע את הצעדים האלה:

צעד 1: הוסף לסרטוט שבמחברתך את היטל הגובה של פונקציית המטרה

בעבור  $z = 5$ .

חשב את שיעורי נקודות החיתוך של היטל זה עם הצירים  $x_1$  ו- $x_2$ ,

וסמן אותן על הסרטוט.

צעד 2: הוסף לסרטוט שבמחברתך את היטל הגובה של פונקציית המטרה

בעבור  $z = 15$ .

חשב את שיעורי נקודות החיתוך של היטל זה עם הצירים  $x_1$  ו- $x_2$ ,

וסמן אותן על הסרטוט.

**צעד 3:** סמן בסרטוט שבמחברתך, באמצעות חץ, את ליוון העלייה של פונקציית

המטרה.

האם הצעדים 1-3 מראים שהפתרון (1, 3) הוא הפתרון האופטימלי היחיד?

נמק את תשובתך.

**ב.** לבעיה הנתונה בתחילת השאלה מוסיפים את האילוץ:

$$x_1 - x_2 \geq 2$$

סרטט במחברתך מערכת צירים חדשה, וסמן עליה רק את תחום הפתרונות האפשריים

שיתקבל לאחר הוספת אילוץ זה.

**ג.** לבעיה הנתונה בתחילת השאלה מוסיפים אילוץ חדש כלשהו, כך שלאחר הוספתו

תחום הפתרונות האפשריים הוא הקטע שבין שתי הנקודות (3, 4) ו-(1, 3).

האם הפתרון האופטימלי הנתון (1, 3) ישתנה? נמק את תשובתך.

**ד.** לבעיה הנתונה בתחילת השאלה מוסיפים אילוץ חדש:

$$3x_1 + 2x_2 \leq 14$$

תלמיד טען שהפתרון האופטימלי הנתון (1, 3) לא ישתנה.

האם התלמיד צדק? נמק את תשובתך בלי לסרטט מחדש את תחום הפתרונות האפשריים

לאחר הוספת אילוץ זה.

**ה.** משנים רק את פונקציית המטרה של הבעיה שבתחילת השאלה ל-  $z = 3x_1$ .

לפניך ההיגדים i-iv שרק אחד מהם נכון. העתק למחברתך את ההיגד הנכון,

ונמק את בחירתך.

**i** הפתרון האופטימלי היחיד יהיה (1, 3).

**ii** הפתרון האופטימלי היחיד יהיה (3, 4).

**iii** הפתרון האופטימלי החדש יהיה:

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \lambda \begin{pmatrix} 3 \\ 1 \end{pmatrix} + (1 - \lambda) \begin{pmatrix} 3 \\ 4 \end{pmatrix}$$

כאשר  $0 \leq \lambda \leq 1$

**iv** הפתרון האופטימלי יהיה לא חסום.

10. בשאלה זו שנישה טעימים, א-ו, שאינם תלויים זה בזה. ענה על כל הטעימים.

א. לפניך בעיית תובלה ובה 2 מקורות ר" 4 יעדים. העלויות ליחידה מכל מקור לכל יעד נתונות בטבלה שלפניך.

מקורות	יעדים				היצע
	1	2	3	4	
1	8	9	4	6	100
2	14	12	13	8	80
ביקוש	10	40	30	70	

על פי הטבלה הנתונה לא ניתן להפעיל את שיטת הפינה הצפון-מערבית כדי למצוא פתרון אפשרי לבעיית התובלה. הסבר מדוע.

ב. בטבלה שלפניך נתון חלק מפתרון בסיסי אפשרי לבעיית תובלה:  $x_{11} = 10$ ,  $x_{12} = 2$ .

מקורות	יעדים			היצע
	1	2	3	
1	4	5	10	12
	10	2		
2	6	3	6	12
3	6	2	6	6
ביקוש	10	10	10	

חצתק את הטבלה למחברתך, והשלם אותה לפי שיטת הפינה הצפון-מערבית.

ג. בטבלה שלפניך נתון פתרון בסיסי אפשרי לבעיית תובלה. הערך שנקבע ל-  $u_1$  הוא 0. העתק את הטבלה למחברתך, חשב את הערך של  $v_1, v_2, v_3, u_2, u_3$  ורשום את הערכים שחישבת במקומות המתאימים בטבלה.

מקורות	יעדים			היצע	$u_i$
	1	2	3		
1	4 10	5 2	10	12	0
2	6	3 2	6 10	12	
3	6	2 6	6	6	
ביקוש	10	10	10		
$v_j$					

(שים לב: המשך השאלה בעמוד הבא.)

/המשך בעמוד 22/

ד. בטבלה שלפניך נתון פתרון בסיסי אפשרי לבעיית תובלה, ונתונים ערפים

של  $u_1, v_1, v_2, v_3, u_1, u_2, u_3$ .

מקורות	יעדים			היצע	$u_j$
	1	2	3		
1	4 20	5 4	10	24	0
2	2	6 16	3 4	20	1
3	6	2	6 8	8	4
ביקוש	20	20	12		
$v_j$	4	5	2		

האם הפתרון הוא אופטימלי? נמק את תשובתך.

ה. בטבלה שלמניך נתון פתרון לבעיית תובלה, ונתונים ערכים של  $v_1, v_2, v_3, u_1, u_2, u_3$ , שמתאימים לפתרון זה.

מקורות	יעדים			היצע	$u_i$
	1	2	3		
1	10 20	25	30	20	10
2	10 30	22	14 20	50	10
3	16	20 40	20 20	60	16
ביקוש	50	40	40		
$v_j$	0	4	4		

לפניך ההיגדים i-iv שרק אחד מהם נכון. העתק למחברתך את ההיגד הנכון, ונמק את בחירתך.

- i הפתרון הנתון איננו פתרון אפשרי.
- ii הפתרון הנתון הוא פתרון בסיסי אפשרי אך לא אופטימלי.
- iii הפתרון הנתון הוא פתרון אופטימלי יחיד.
- iv הפתרון הנתון הוא פתרון אופטימלי אך איננו פתרון אופטימלי יחיד.

(שים לב: המשך השאלה בעמוד הבא.)

ו. בטבלה שלפניך נתון חלק מפתרון בסיסי אפשרי לבעיית תובלה שחושב על פי שיטת הפינה הצפון מערבית, ונתונים ערכים של  $u_1, u_2, u_3, v_1, v_2, v_3$  שחושבו על פי פתרון זה.

מקורות	יעדים			היצע	$u_i$
	1	2	3		
1	12 20	15	17	20	2
2	10	18 10	14	10	0
3	20	10 5	18 10	15	-8
ביקוש	20	15	10		
$v_j$	10	18	26		

מה צריך להוסיף לטבלה כדי לקבל פתרון בסיסי אפשרי המתאים לכל הנתונים שבטבלה?

העתק למחברתך את התשובה הנכונה מבין האפשרויות i-iv, ונמק את תשובתך.

.  $X_{12} = 0$     i

.  $X_{21} = 0$     ii

.  $X_{13} = 0$     iii

$X_{31} = 0$     iv



מודלים חישוביים

אם למדת מסלול זה, ענה על אחת מהשאלות 11-12 (25 נקודות).

11. לפניך השפה  $L$  מעל הא"ב  $\{0, \$\}$ :

$$L = \left\{ \begin{array}{l} 0^3 \$ 0^i \$ 0^{i-2} \$ \dots 0^i \$ \\ k \geq 1 \\ \text{לכל } m \text{ בין } 1 \text{ ל- } k: i_m \geq 0 \\ \text{ו- } i_m \text{ מתחלק ב- } 3 \text{ ללא שארית} \end{array} \right.$$

א. כתוב את המילה הקצרה ביותר בשפה  $L$ .

ב. בנה אוטומט סופי דטרמיניסטי שיקבל את השפה  $L$ .

/המשך בעמוד 26/

12. נבנתה מכונת טיורינג המחשבת את הפונקציה  $f(m, n)$ . מכונה זו מוצאת את המספר הקטן מבין שני מספרים  $m$  ו- $n$ . המכונה מקבלת כקלט שני מספרים  $m$  ו- $n$  שלמים וגדולים מ-0. שני המספרים רשומים על הסרט כמספרים אונריים (מספר אונרי  $m$  הוא מספר המיוצג על ידי  $m$  תווים של 1) ובין שני המספרים השום הסימן #. הפלט של המכונה הוא המספר הקטן מבין שני המספרים, והוא יירשם על הסרט כמספר אונרי בין שני סימני \$. סימן ה-\$ השמאלי יירשם במקום הסימן #, והמספר יירשם מימינו. במהלך הפעולה המכונה יכולה להיעזר בסימנים  $a, b$ . לדוגמה: בעבור הקלט  $m = 3$  ו- $n = 5$ , לפני החישוב סרט הזיכרון ייראה כך:

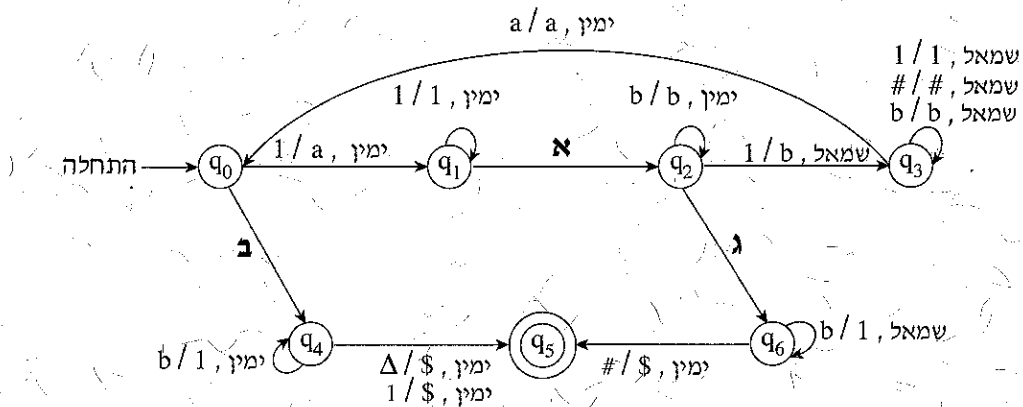
F	1	1	1	#	1	1	1	1	1	Δ	Δ	Δ	...
---	---	---	---	---	---	---	---	---	---	---	---	---	-----

לאחר סיום החישוב ייראה הסרט כך:

F	...	...	...	\$	1	1	1	\$	...
---	-----	-----	-----	----	---	---	---	----	-----

הערות: התאים המסומנים ב-... הם תאים שאין חשיבות לתוכנם.

לפניך סרטוט חלקי של המכונה.



א. בסרטוט יש שלושה מעברים המסומנים באותיות א-ג.

במעברים אלה חסרים סימני הקלט וההוראות.

העתיק למחברתך את הסרטוט, והשלם את שלושת המעברים החסרים כך שהמכונה תחשב

את הפונקציה  $f(m, n)$ .

ב. הראה את תהליך החישוב של המכונה בעבור הקלט  $n = 1, m = 1$ .

בכל שלב רשום את מצב הסרט, סמן היכן נמצא ראש המכונה, ורשום באיזה מהמצבים

$q_6 - q_0$  המכונה נמצאת.

**תכנות מונחה עצמים**

אם למדת מסלול זה ואתה כותב ב-Java, ענה על אחת מהשאלות 13-14. (25 נקודות)  
 13. לפניך חלק מפרויקט העוסק בכלי תחבורה וכולל את המחלקות האלה:

המחלקה Vehicle	מייצגת כלי תחבורה
המחלקה Train	מייצגת רכבת שהיא כלי תחבורה
המחלקה Boat	מייצגת סירה שהיא כלי תחבורה
המחלקה Airplane	מייצגת מטוס שהיא כלי תחבורה
המחלקה TransportationCompany	מייצגת חברה שיש לה כלי תחבורה מסוגים שונים

**Public class Vehicle**

```

{
    private String type;           // מיקום (יבשה / מים / אוויר)
    private String way;           // סוג הדרך (כביש / מסילה / נהר / ...)
    private int maxSpeed          // מהירות מקסימלית

    public Vehicle(String type, String way, int maxSpeed)
    {
        this.type = type;
        this.way = way;
        this.maxSpeed = maxSpeed
    }
}
    
```

**public class Train extends Vehicle**

```

{
    private int numOfCarriages    // מספר הקרונות
    public Train(int maxSpeed, int numOfCarriages)
    {
        super("land", "tracks", maxSpeed);
        this.numOfCarriages = numOfCarriages;
    }
    public void incNumOfCarriages(int n) // מגדילה ב-n את מספר הקרונות ברכבת
    {
        this.numOfCarriages = this.numOfCarriages + n;
    }
}
    
```

```
public class Boat extends Vehicle
```

```
{
    public Boat(String way, int maxSpeed)
    {
        super("water", way , maxSpeed);
    }
}
```

```
public class Airplane extends Vehicle
```

```
{
    private int maxHeight; // גובה טיסה מקסימלי
    public Airplane(int maxSpeed, int maxHeight)
    {
        super("sky", "air", maxSpeed)
        this.maxHeight = maxHeight;
    }
}
```

```
public class TransportationCompany
```

```
{
    private Vehicle[] vehicles = new Vehicle[50]; // מערך כלי התחבורה בחברה
    private int counter = 0; // מספר כלי התחבורה שיש בפועל
    public TransportationCompany()
    {
    }

    public void addVehicle (Vehicle v) // מוסיפה כלי תחבורה למערך כלי התחבורה של
    { // החברה, הנח שיש מקום להוסיף כלי תחבורה.
        this.vehicles[counter] = v;
        this.counter++;
    }
}
```

(שים לב: המשך השאלה, בעמוד הבא.)

א. ממש ב־ Java מחלקה ראשית Program ובה פעולה ראשית, שתבצע את המשימות האלה:

i בנייה של עצם מטיפוס חברה של כלי תחבורה — **TransportationCompany**

הנקרא company1.

ii הוספה של סירה אחת ורכבת אחת לחברה company1.

בחר לתכונות ערכים כרצונך.

ב. במחלקה **TransportationCompany** הוגדרה הפעולה:

```
public void display()
{
    for (int i=0; i<this.counter; i++)
    {
        System.out.println((i+1) + ":" + this.vehicles[i]);
    }
}
```

ממש ב־ Java פעולות שיאפשרו ביצוע תקין של הפעולה display(), כך שבעבור כל כלי תחבורה יודפסו כל התכונות שלו. הגדר את הפעולות באופן המתאים ביותר לעקרונות של תכנות מונחה עצמים (הכמסה — encapsulation, הורשה — inheritance, פולימורפיזם — polymorphism).

בעבור כל פעולה שאתה מממש, רשום לאיזו מחלקה היא שייכת.

אן לשנות את הפעולה display().

ג. ממש ב־ Java פעולה שתקבל מספר שלם n ותוסיף n קרונות לכל הרכבות ששייכות

לחברה שיש לה כלי תחבורה מסוגים שונים, תעד את הפעולה, ורשום באיזו מחלקה יש

להגדיר אותה. אן לשנות את הפעולות הקיימות בפרויקט.

14. לפניך המחלקות AA ו- BB :

```

public class AA
{
    private String st;

    public AA() { this.st = "excellent"; }
    public AA(String st) { this.st = st; }
    public String getSt() { return this.st; }
    public void setSt(String st) { this.st = st; }
    public String toString() { return "st = " + this.st; }
}

public class BB extends AA
{
    private int num;

    public BB() { super(); this.num = 1; }
    public BB(int num , String st) { super(st); this.num = Math.abs(num); }
    public int getNum() { return this.num; }
    public void setNum(int num) { this.num = num; }
    public String toString() { return super.toString() + " num = " + this.num; }
}
    
```

- א. הגדר במחלקה AA פעולה בוליאנית בשם isLike(Object obj) המקבלת עצם obj מטיפוס Object. אם העצם obj הינו מטיפוס AA וגם תוכן המחרוזת st של obj זהה לתוכן המחרוזת st של העצם הנוכחי — הפעולה תחזיר true, אחרת — תחזיר false.
- ב. הגדר במחלקה BB פעולה הדורסת את הפעולה שהגדרת בסעיף א.
- אם העצם obj הינו מטיפוס BB וגם ערך התכונה num שלו זהה לערך התכונה num של העצם הנוכחי — הפעולה תחזיר true, אחרת — תחזיר false.

ג. לפניך קטע מפעולה ראשית:

```
AA a = new AA("excellent");
BB b = new BB();
a = b;
if (a.isLike(b)) System.out.println(a);
```

האם קטע התכנית תקיין?

אם כן — מה יהיה פלט הקטע? רשום איזו גרסה של הפעולה isLike תופעל — זו של AA או זו של BB.

אם לא — הסבר מהי השגיאה ומתי היא תתגלה: בזמן קומפילציה או בזמן ריצה.

ד. לפניך קטע מפעולה ראשית:

```
AA aa = new AA();
BB bb = new BB(2, "excellent");
bb = aa;
if (bb.isLike(aa)) System.out.println(bb);
```

האם קטע התכנית תקיין?

אם כן — מה יהיה פלט הקטע? רשום איזו גרסה של הפעולה isLike תופעל — זו של AA או זו של BB.

אם לא — הסבר מהי השגיאה ומתי היא תתגלה: בזמן קומפילציה או בזמן ריצה.

ה. כתוב פעולה חיצונית בשם longString המקבלת מערך של עצמים מטיפוס Object.

הפעולה מחזירה מחרוזת המורכבת משרשר התכונה st של עצמים מטיפוס AA במערך, באופן הזה:

— אם לעצם יש רק התכונה st, תשרשר המחרוזת שבתכונה st פעם אחת.

— אם לעצם יש גם התכונה num, המחרוזת שבתכונה st תשרשר num פעמים.

— אם אין במערך אף עצם מטיפוס AA, תוחזר מחרוזת ריקה.

**תכנות מונחה עצמים**

15. אם למדת מסלול זה ואתה כותב ב- C#, ענה על אחת מהשאלות 15-16. (25 נקודות)

15. לפניך חלק מפרויקט העוסק בכלי תחבורה וכולל את המחלקות האלה:

המחלקה Vehicle	מייצגת כלי תחבורה
המחלקה Train	מייצגת רכבת שהיא כלי תחבורה
המחלקה Boat	מייצגת סירה שהיא כלי תחבורה
המחלקה Airplane	מייצגת מטוס שהיא כלי תחבורה
המחלקה TransportationCompany	מייצגת חברה שיש לה כלי תחבורה מסוגים שונים

**Public class Vehicle**

```
{
    private string type;           // מיקום (יבשה / מים / אוויר)
    private string way;           // סוג הדרך (כביש / מסילה / נהר / ...)
    private int maxSpeed          // מהירות מקסימלית

    public Vehicle(string type, string way, int maxSpeed)
    {
        this.type = type;
        this.way = way;
        this.maxSpeed = maxSpeed
    }
}
```

**public class Train : Vehicle**

```
{
    private int numOfCarriages    // מספר הקרונות
    public Train(int maxSpeed, int numOfCarriages) : base("land", "tracks", maxSpeed)
    {
        this.numOfCarriages = numOfCarriages;
    }
    public void IncNumOfCarriages(int n) // מגדילה ב-n את מספר הקרונות ברכבת
    {
        this.numOfCarriages = this.numOfCarriages + n;
    }
}
```

(שים לב: המשך השאלה בעמודים הבאים.)



public class **Boat : Vehicle**

```
{
    public Boat(string way, int maxSpeed) : base("water", way, maxSpeed)
    {
    }
}
```

public class **Airplane : Vehicle**

```
{
    private int maxHeight; // טובה טיסה מקסימלי
    public Airplane(int maxSpeed, int maxHeight) : base("sky", "air", maxSpeed)
    {
        this.maxHeight = maxHeight;
    }
}
```

public class **TransportationCompany**

```
{
    private Vehicle[] vehicles = new Vehicle[50]; // מערך כלי התחבורה בחברה
    private int counter = 0; // מספר כלי התחבורה שיש בפועל
    public TransportationCompany()
    {
    }

    public void AddVehicle (Vehicle v) // מוסיפה כלי תחבורה למערך כלי התחבורה של
    { // החברה. הנח שיש מקום להוסיף כלי תחבורה.
        this.vehicles[counter] = v;
        this.counter++;
    }
}
```

(שים לב: המשך השאלה בעמוד הבא.)

א. ממש ב- C# מחלקה ראשית Program ובה פעולה ראשית, שתבצע את המשימות האלה:

- i בנייה של עצם מטיפוס חברה של כלי תחבורה – **TransportationCompany** הנקרא company1.
- ii הוספה של סירה אחת ורכבת אחת לתברה company1. בחר לתכונות ערכים כרצונך.

ב. במחלקה **TransportationCompany** הוגדרה הפעולה:

```
public void Display()
{
    for (int i=0; i<this.counter; i++)
    {
        Console.WriteLine((i+1) + ":" + this.vehicles[i]);
    }
}
```

ממש ב- C# פעולות שיאפשרו ביצוע תקין של הפעולה Display(), כך שבעבור כל כלי תחבורה יודפסו כל התכונות שלו. הגדר את הפעולות באופן המתאים ביותר לעקרונות של תכנות מונחה עצמים (הכמסה – encapsulation, הורשה – inheritance, פולימורפיזם – polymorphism), בעבור כל פעולה שאתה מממש, רשום לאיזו מחלקה היא שייכת. אץ לשנות את הפעולה Display().

ג. ממש ב- C# פעולה, שתקבל מספר שלם n ותוסיף n קורונות לכל הרכבות ששייכות לחברה שיש לה כלי תחבורה מסוגים שונים. תעד את הפעולה, ורשום באיזו מחלקה יש להגדיר אותה. אץ לשנות את הפעולות הקיימות בפרויקט.

16. לפיך המחלקות AA ו- BB:

```
public class AA
{
    private string st;

    public AA() { this.st = "excellent"; }
    public AA(string st) { this.st = st; }
    public string GetSt() { return this.st; }
    public void SetSt (string st) { this.st = st; }
    public override string ToString() { return "st = " + this.st; }
}
```

```
public class BB : AA
{
    private int num;

    public BB() : base() { this.num = 1; }
    public BB(int num, string st) : base(st) { this.num = Math.Abs(num); }
    public int GetNum() { return this.num; }
    public void SetNum(int num) { this.num = num;}
    public override string ToString() { return base.ToString() + " num = "
        + this.num;}
}
```

א. הגדר במחלקה AA פעולה בוליאנית הניתנת לדריסה, בשם (Object obj) IsLike, המקבלת עצם obj מטיפוס Object. אם העצם obj הינו מטיפוס AA וגם תוכן המחרוזת st של obj זהה לתוכן המחרוזת st של העצם הנוכחי – הפעולה תחזיר true, אחרת – תחזיר false.

ב. הגדר במחלקה BB פעולה הדורסת את הפעולה שהגדרת בסעיף א.

אם העצם obj הינו מטיפוס BB וגם ערך התכונה num שלו זהה לערך התכונה num של העצם הנוכחי – הפעולה תחזיר true/אחרת – תחזיר false.

(שים לב: המשך השאלה בעמוד הבא.)

ג. לפניך קטע מפעולה ראשית:

```
AA a = new AA("excellent");
BB b = new BB();
a = b;
if (a.IsLike(b)) Console.WriteLine(a);
```

האם קטע התכנית תקיין?

אם כן – מה יהיה פלט הקטע? רשום איזו גרסה של הפעולה IsLike תופעל – זו של AA או זו של BB.

אם לא – הסבר מהי השגיאה ומתי היא תתגלה: בזמן קומפילציה או בזמן ריצה.

ד. לפניך קטע מפעולה ראשית:

```
AA aa = new AA();
BB bb = new BB(2, "excellent");
bb = aa;
if (bb.IsLike(aa)) Console.WriteLine(bb);
```

האם קטע התכנית תקיין?

אם כן – מה יהיה פלט הקטע? רשום איזו גרסה של הפעולה IsLike תופעל – זו של AA או זו של BB.

אם לא – הסבר מהי השגיאה ומתי היא תתגלה: בזמן קומפילציה או בזמן ריצה.

ה. כתוב פעולה חיצונית בשם LongString המקבלת מערך של עצמים מטיפוס Object.

הפעולה מחזירה מחרוזת המורכבת משרשרת התכונה st של עצמים מטיפוס AA במערך, באופן הזה:

— אם לעצם יש בן התכונה st, תשרשר המחרוזת שבתכונה st פעם אחת.

→ אם לעצם יש גם התכונה num, המחרוזת שבתכונה st תשרשר num פעמים.

— אם אין במערך אף עצם מטיפוס AA, תוחזר מחרוזת ריקה.

## בהצלחה!