

שים לב: בבחינה זו יש הנחיות מיוחדות.  
יש לענות על השאלות על פי הנחיות אלה.

## מדעי המחשב

### הוראות לנבחן

- משך הבחינה: שלוש שעות.
- מבנה השאלון ומפתח ההערכה: בשאלון זה שלושה פרקים. יש לבחור בחמש שאלות מתוך שלושת הפרקים. לכל שאלה — 20 נקודות ( $20 \times 5$ ). סך הכול — 100 נקודות. שים לב: אם תבחר לענות על שאלות מן הפרק השלישי, בחר בשאלות מתוך מסלול אחד בלבד.
- חומר עזר מותר בשימוש: כל חומר עזר, חוץ ממחשבון שיש בו אפשרות תכנות.
- הוראות מיוחדות:
  - רשום על הכריכה החיצונית של המחברת את שם המסלול שלמדת, אם תבחר לענות על שאלות מן הפרק השלישי, ואם לא — רשום "ללא מסלול". המסלול הוא אחד מארבעת המסלולים האלה: מערכות מחשב ואסמבלי, מבוא לחקר ביצועים, מודלים חישוביים, תכנות מונחה עצמים.
  - את כל התוכניות שאתה נדרש לכתוב בשפת מחשב בפרקים הראשון והשני כתוב בשפה אחת בלבד — Java או C#.הערה: לא יורדו לך נקודות אם בתוכניות שאתה כותב תכתוב אות גדולה במקום אות קטנה או להפך.

כתוב במחברת הבחינה בלבד. רשום "טיוטה" בראש כל עמוד המשמש טיוטה.  
כתובת טיוטה בדפים שאינם במחברת הבחינה עלולה לגרום לפסילת הבחינה.

ההנחיות בשאלון זה מנוסחות בלשון זכר ומכוונות לנבחנות ולנבחנים כאחד.

בהצלחה!

**שים לב: השאלות מתחילות בעמוד 3.**

## השאלות

### בשאלון זה שלושה פרקים.

יש לבחור בחמש שאלות מתוך שלושת הפרקים. לכל שאלה – 20 נקודות.  
רשום על הכריכה החיצונית של המחברת את שם המסלול שלמדת, אם תבחר לענות  
על שאלות מן הפרק השלישי, ואם לא – רשום "ללא מסלול".

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

לפותרים בשפת Java: בכל שאלה שנדרשת בה קליטה, הנח שבתוכנית כתובה ההוראה:

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

## פרק ראשון

1. כתוב פעולה חיצונית ששמה filter בשפת Java או Filter בשפת C# המקבלת מערך מטיפוס שלם – arr ומספר מטיפוס שלם – num. על הפעולה להחזיר מערך חדש מטיפוס שלם, הכולל רק את המספרים המופיעים במערך arr שאינם שווים ל- num.

לדוגמה: בעבור num ששווה ל-9 ומערך arr בגודל 7 שלפניך:

	0	1	2	3	4	5	6
arr	6	9	2	2	9	4	-3

הפעולה תחזיר מערך בגודל 5, הנראה כך:

	0	1	2	3	4
	6	2	2	4	-3

הנח שקיים במערך arr מספר אחד לפחות שאיננו שווה ל- num ומספר אחד לפחות ששווה ל- num.

שים לב: גודל המערך המוחזר הוא כמספר האיברים במערך arr שאינם שווים ל- num.

הערה: אין חשיבות לסדר המספרים במערך המוחזר.

## 2. שים לב! לשאלה זו שני נוסחים: בשפת Java, בעמודים 4-5 ובשפת C# בעמודים 6-7.

### לפותרים בשפת Java

נתונה המחלקה **Subject**, המייצגת מקצוע בתעודה של תלמיד, ולה שתי תכונות: שם המקצוע — `subName` והציון — `grade` (בין 0 ל-100).

```
public class Subject {
    private String subName;
    private int grade;
}
```

נוסף על כך, נתונה המחלקה **ReportCard**, המייצגת תעודה של תלמיד, ולה שתי תכונות: שם התלמיד — `stuName` ומערך של עצמים — `subArray` מטיפוס **Subject** בגודל מספר המקצועות שלמד התלמיד.

```
public class ReportCard {
    private String stuName;
    private Subject [] subArray;
}
```

הנח שקיימות פעולות `get` ו-`set` בעבור כל התכונות בשתי המחלקות.

לפניך ממשק חלקי של המחלקה **ReportCard**.

כותרת הפעולה	תיאור הפעולה
<code>public ReportCard (String name, int num)</code>	פעולה בונה המקבלת את שם התלמיד ואת מספר המקצועות שלמד התלמיד, ומאתחלת את תכונות המחלקה. המערך <code>subArray</code> מאותחל למערך ריק בגודל <code>num</code> .
<code>public double average ()</code>	פעולה המחזירה את ממוצע כל הציונים בתעודה של התלמיד.
<code>public boolean isExcellent ()</code>	פעולה המחזירה <code>true</code> אם התלמיד נחשב לתלמיד מצטיין, אחרת היא מחזירה <code>false</code> . תלמיד מצטיין: תלמיד שממוצע ציוניו הוא 85 ומעלה, כל ציוניו מעל 54, ולפחות אחד מציוניו הוא 100.

(שים לב: המשך השאלה בעמוד הבא.)

- א. ממש את הפעולה הבונה של המחלקה **ReportCard**.
- ב. ממש את הפעולה `isExcellent` במחלקה **ReportCard**.  
אפשר להשתמש בפעולה `average` של המחלקה **ReportCard** בלי לממש אותה.  
הנח שאיברי המערך `subArray` שונים מ-`null`.
- ג. כתוב פעולה חיצונית ששמה `printExcellent` המקבלת מערך של עצמים — `array` מטיפוס **ReportCard**, המייצג תלמידים בכיתה מסוימת. הפעולה תדפיס את שמות התלמידים המצטיינים.  
חובה להשתמש בפעולה `isExcellent` של המחלקה **ReportCard**.  
הנח שאיברי המערך `array` שונים מ-`null`.

### לפותרים בשפת C#

נתונה המחלקה **Subject**, המייצגת מקצוע בתעודה של תלמיד, ולה שתי תכונות: שם המקצוע — subName והציון — grade (בין 0 ל-100).

```
public class Subject {
    private string subName;
    private int grade;
}
```

נוסף על כך, נתונה המחלקה **ReportCard**, המייצגת תעודה של תלמיד, ולה שתי תכונות: שם התלמיד — stuName ומערך של עצמים — subArray מטיפוס **Subject** בגודל מספר המקצועות שלמד התלמיד.

```
public class ReportCard {
    private string stuName;
    private Subject [] subArray;
}
```

הנח שקיימות פעולות Get ו- Set בעבור כל התכונות בשתי המחלקות.

לפניך ממשק חלקי של המחלקה **ReportCard**.

כותרת הפעולה	תיאור הפעולה
public ReportCard (string name, int num)	פעולה בונה המקבלת את שם התלמיד ואת מספר המקצועות שלמד התלמיד, ומאתחלת את תכונות המחלקה. המערך subArray מאותחל למערך ריק בגודל num.
public double Average ()	פעולה המחזירה את ממוצע כל הציונים בתעודה של התלמיד.
public bool IsExcellent ()	פעולה המחזירה true אם התלמיד נחשב לתלמיד מצטיין, אחרת היא מחזירה false. תלמיד מצטיין: תלמיד שממוצע ציוניו הוא 85 ומעלה, כל ציוניו מעל 54, ולפחות אחד מציוניו הוא 100.

(שים לב: המשך השאלה בעמוד הבא.)

- א. ממש את הפעולה הבונה של המחלקה **ReportCard**.
- ב. ממש את הפעולה **IsExcellent** במחלקה **ReportCard**.  
אפשר להשתמש בפעולה **Average** של המחלקה **ReportCard** בלי לממש אותה.  
הנח שאיברי המערך **subArray** שונים מ-**null**.
- ג. כתוב פעולה חיצונית ששמה **PrintExcellent** המקבלת מערך של עצמים — **array** מטיפוס **ReportCard**, המייצג תלמידים בכיתה מסוימת. הפעולה תדפיס את שמות התלמידים המצטיינים.  
חובה להשתמש בפעולה **IsExcellent** של המחלקה **ReportCard**.  
הנח שאיברי המערך **array** שונים מ-**null**.

## 3. שים לב! לשאלה זו שני נוסחים: בשפת Java בעמודים 8-9 ובשפת C# בעמודים 10-11.

לפותרים בשפת Java

"מחרוזת מיוחדת" היא מחרוזת שבה התווים הזהים מופיעים ברצף או מחרוזת ללא תווים זהים (כולל מחרוזת ריקה).

דוגמאות ל"מחרוזת מיוחדת": "1zz\*\*", "&\$\*", "a", "aaaa", "ccabbb".

דוגמאות למחרוזות שאינן "מחרוזת מיוחדת": "3%3\*\*", "aabcba", "aba".

נתונה המחלקה **MyString**.

```
public class MyString {
    private String str;
}
```

לפניך ממשיך המחלקה **MyString**.

יש להשתמש בפעולות הממשק ללא צורך לממש אותן.

שם הפעולה	כותרת הפעולה
פעולה הבונה מחרוזת ריקה.	public MyString ()
פעולה המחזירה את מספר המופעים של התו ch במחרוזת str.	public int countChar (char ch)
פעולה המקבלת תו ch ומוחקת מן המחרוזת str את כל המופעים שלו.	public void removeChar (char ch)
פעולה המוסיפה לסוף המחרוזת str את התו ch.	public void appendChar (char ch)
פעולה המחזירה את התו הראשון במחרוזת str.	public char firstChar ()
פעולה המחזירה true אם המחרוזת str ריקה, אחרת היא מחזירה false.	public boolean isEmpty ()

(שים לב: המשך השאלה בעמוד הבא.)



לפניך כותרת של הפעולה החיצונית `special`:

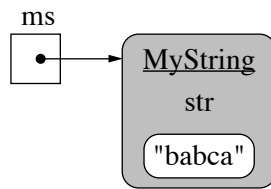
```
public static MyString special (MyString ms)
```

הפעולה `special` מחזירה עצם חדש מטיפוס **MyString** שהתכונה שלו, `str`, היא "מחרוזת מיוחדת" המורכבת מכל התווים שבתכונה `str` של העצם `ms`.

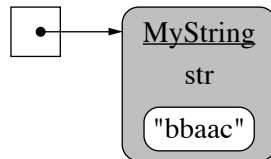
ממש את הפעולה `special` באמצעות פעולות ממשק המחלקה **MyString** בלבד. אנן להוסיף פעולות, גם לא פעולות `get` ו-`set`, למחלקה **MyString**.

לדוגמה:

בעבור העצם `ms` שלפניך, שבו מחרוזת רגילה:



הפעולה `special` תחזיר עצם חדש, שבו "מחרוזת מיוחדת":



הערות:

- אין צורך לשמור על המחרוזת `str` בעצם `ms`.
- אין חשיבות לסדר התווים בעצם החדש, כל עוד נשמרת ההגדרה של "מחרוזת מיוחדת".

## לפותרים בשפת C#

"מחרוזת מיוחדת" היא מחרוזת שבה התווים הזהים מופיעים ברצף או מחרוזת ללא תווים זהים (כולל מחרוזת ריקה).

דוגמאות ל"מחרוזת מיוחדת": "1zz\*\*", "&\$\*", "a", "aaaa", "ccabbb".

דוגמאות למחרוזות שאינן "מחרוזת מיוחדת": "33%3\*\*", "abcb", "aba".

נתונה המחלקה **MyString**.

```
public class MyString {
    private string str;
}
```

לפניך ממשק המחלקה **MyString**.

יש להשתמש בפעולות הממשק ללא צורך לממש אותן.

שם הפעולה	כותרת הפעולה
פעולה הבונה מחרוזת ריקה.	public MyString ()
פעולה המחזירה את מספר המופעים של התו ch במחרוזת str.	public int CountChar (char ch)
פעולה המקבלת תו ch ומוחקת מן המחרוזת str את כל המופעים שלו.	public void RemoveChar (char ch)
פעולה המוסיפה לסוף המחרוזת str את התו ch.	public void AppendChar (char ch)
פעולה המחזירה את התו הראשון במחרוזת str.	public char FirstChar ()
פעולה המחזירה true אם המחרוזת str ריקה, אחרת היא מחזירה false.	public bool IsEmpty ()

(שים לב: המשך השאלה בעמוד הבא.)

לפניך כותרת של הפעולה החיצונית Special:

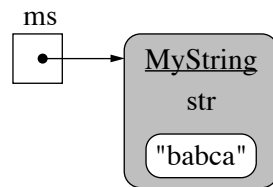
public static MyString Special (MyString ms)

הפעולה Special מחזירה עצם חדש מטיפוס **MyString** שהתכונה שלו, str, היא "מחרוזת מיוחדת" המורכבת מכל התווים שבתכונה str של העצם ms.

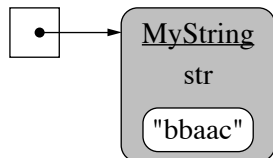
ממש את הפעולה Special באמצעות פעולות ממשק המחלקה **MyString** בלבד. אין להוסיף פעולות, גם לא פעולות Get ו-Set, למחלקה **MyString**.

לדוגמה:

בעבור העצם ms שלפניך, שבו מחרוזת רגילה:



הפעולה Special תחזיר עצם חדש, שבו "מחרוזת מיוחדת":



הערות:

- אין צורך לשמור על המחרוזת str בעצם ms.
- אין חשיבות לסדר התווים בעצם החדש, כל עוד נשמרת ההגדרה של "מחרוזת מיוחדת".

## פרק שני

**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.

4. בשאלה זו תוכל להשתמש בפעולה החיצונית שלפניך בלי לממש אותה.

תיאור הפעולה	כותרת הפעולה
<p>הפעולה מקבלת מספר — num והפניה לתחילת שרשרת חוליות — lst. הפעולה מוחקת את החוליות שבהן הערך num ומחזירה הפניה לתחילת שרשרת החוליות.</p>	<p><b>בשפת Java</b>  <code>public static Node&lt;Integer&gt; delete (int num, Node&lt;Integer&gt; lst)</code></p> <p><b>בשפת C#</b>  <code>public static Node&lt;int&gt; Delete (int num, Node&lt;int&gt; lst)</code></p>

נתונה המחלקה **BiList** — דו־שרשרת, ולה שתי תכונות:

- lst1 — הפניה לתחילת שרשרת חוליות מטיפוס שלם
- lst2 — הפניה לתחילת שרשרת חוליות מטיפוס שלם

לפניך ממשק חלקי של המחלקה **BiList** בשפות **Java** ו־**C#**.

יש להשתמש בפעולות הממשק ללא צורך לממש אותן.

תיאור הפעולה	כותרת הפעולה
<p>פעולה הבונה את העצם עם הפניות לשתי שרשראות ריקות.</p>	<p><code>public BiList ()</code></p>
<p>פעולה המוסיפה חוליה שבה הערך num לסוף השרשרת lst1 או לסוף השרשרת lst2 בהתאם ל־ codeList:  כאשר codeList = 1, יוכנס num ל־ lst1,  וכאשר codeList = 2, יוכנס num ל־ lst2.  הנח שהערך של הפרמטר codeList תקין.</p>	<p><b>בשפת Java</b>  <code>public void addNum (int num, int codeList)</code></p> <p><b>בשפת C#</b>  <code>public void AddNum (int num, int codeList)</code></p>

(שים לב: המשך השאלה בעמוד הבא.)

כתוב פעולה חיצונית ששמה generateBilist בשפת Java או GenerateBilist בשפת C# המקבלת שרשרת חוליות — lst של מספרים שלמים. מספר החוליות ב־ lst זוגי והמספרים בחוליות שלה שונים זה מזה.

הפעולה תחזיר עצם מטיפוס **BiList** שמתקיימים בו התנאים האלה:

— כל אחד מן המספרים שבשרשרת lst יופיע באחת מן השרשראות lst1 ו־ lst2.

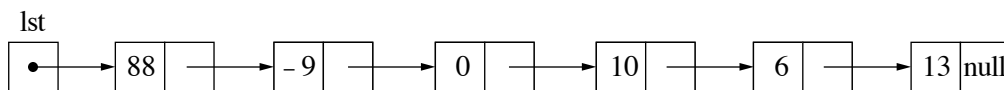
— כל המספרים בשרשרת lst1 יהיו גדולים מכל המספרים בשרשרת lst2.

— מספר החוליות בשתי השרשראות lst1 ו־ lst2 יהיה זהה.

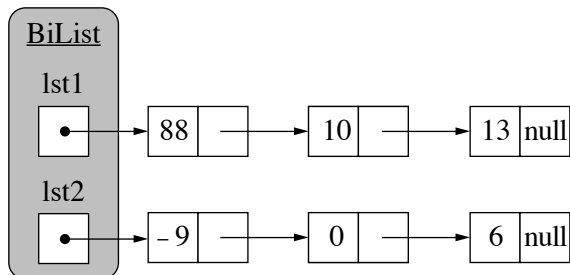
שים לב: אין להוסיף פעולות, גם לא פעולות get ו־ set בשפת Java או Get ו־ Set בשפת C#, למחלקה **BiList**.

דוגמה:

בעבור השרשרת lst שלפניך:



הפעולה תחזיר את העצם הזה:



הערות:

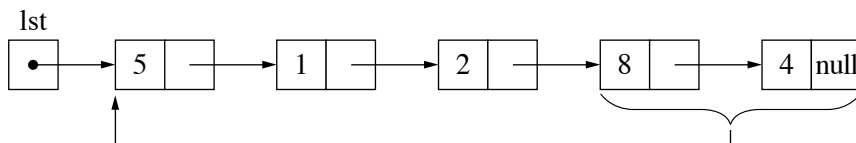
— אין צורך לשמור על השרשרת lst.

— אין חשיבות לסדר האיברים בשרשרת lst1 ובשרשרת lst2.

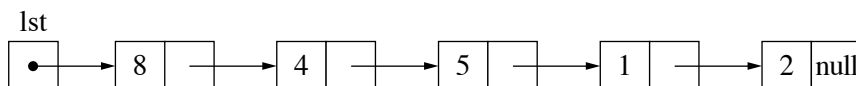
5. בשרשרת חוליות, "העברה מעגלית של  $n$  חוליות" היא העברת  $n$  החוליות האחרונות לתחילת השרשרת (בלי לשנות את סדר הופעתן).

בדוגמה שלפניך בעבור  $n = 2$  : מעבירים את שתי החוליות האחרונות לתחילת השרשרת.

שרשרת החוליות לפני ההעברה



שרשרת החוליות לאחר ההעברה



- א. כתוב פעולה חיצונית ששמה `move` בשפת `Java` או `Move` בשפת `C#` המקבלת שרשרת חוליות `lst` מטיפוס שלם ומספר  $n$  מטיפוס שלם. הפעולה תחזיר את שרשרת החוליות לאחר "העברה מעגלית של  $n$  חוליות".  
הנח:  $n \geq 0$  ומספר החוליות בשרשרת גדול מ- $n$ .
- ב. מהי סיבוכיות זמן הריצה של הפעולה שכתבת בסעיף א? נמק.

6. שים לב! לשאלה זו שני נוסחים: בשפת Java בעמוד 15 ובשפת C# בעמוד 16.

בשפת Java

א. לפניך הפעולה sod1.

```
public static boolean sod1 (int[] arr, int x, int i)
{
    if ( i == -1) return false;
    if (arr[i] == x) return true;
    return sod1(arr, x, i - 1);
}
```

(1) כתוב את הערך המוחזר בעבור זימון הפעולה sod1(a, 8, a.length - 1) והמערך a שלפניך.

עליך להראות מעקב.

	0	1	2	3	4
a	5	4	15	12	2

(2) בעבור מערך כלשהו a ומספר x, מהי מטרת הפעולה sod1(a, x, a.length - 1) ?

(3) מהי סיבוכיות זמן הריצה של הפעולה sod1 ? נמק.

ב. לפניך הפעולה sod2.

```
public static boolean sod2 (int[] arr, int x, int i)
{
    if ( i == 0) return false;
    if (sod1(arr, x - arr[i], i - 1)) return true;
    return sod2(arr, x, i - 1);
}
```

(1) כתוב את הערך המוחזר בעבור זימון הפעולה sod2(a, 16, a.length - 1) והמערך a שלפניך.

עליך להראות מעקב.

	0	1	2	3	4
a	5	4	15	12	2

בסעיף זה, אין צורך לבצע מעקב אחר הפעולה sod1.

(2) בעבור מערך כלשהו a ומספר x, מהי מטרת הפעולה sod2(a, x, a.length - 1) ?

(3) מהי סיבוכיות זמן הריצה של הפעולה sod2 ? נמק.

בשפת C#

א. לפניך הפעולה Sod1.

```
public static bool Sod1 (int[] arr, int x, int i)
{
    if ( i == -1) return false;
    if (arr[i] == x) return true;
    return Sod1(arr, x, i - 1);
}
```

- (1) כתוב את הערך המוחזר בעבור זימון הפעולה Sod1(a, 8, a.Length - 1) והמערך a שלפניך. עליך להראות מעקב.

	0	1	2	3	4
a	5	4	15	12	2

- (2) בעבור מערך כלשהו a ומספר x, מהי מטרת הפעולה Sod1(a, x, a.Length - 1)?
- (3) מהי סיבוכיות זמן הריצה של הפעולה Sod1? נמק.

ב. לפניך הפעולה Sod2.

```
public static bool Sod2 (int[] arr, int x, int i)
{
    if ( i == 0) return false;
    if (Sod1(arr, x - arr[i], i - 1)) return true;
    return Sod2(arr, x, i - 1);
}
```

- (1) כתוב את הערך המוחזר בעבור זימון הפעולה Sod2(a, 16, a.Length - 1) והמערך a שלפניך. עליך להראות מעקב.

	0	1	2	3	4
a	5	4	15	12	2

- בסעיף זה, אין צורך לבצע מעקב אחר הפעולה Sod1.
- (2) בעבור מערך כלשהו a ומספר x, מהי מטרת הפעולה Sod2(a, x, a.Length - 1)?
- (3) מהי סיבוכיות זמן הריצה של הפעולה Sod2? נמק.

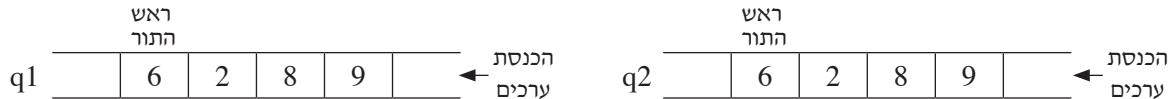


7. בשאלה זו תוכל להשתמש בפעולה החיצונית שלפניך בלי לממש אותה.

כותרת הפעולה	תיאור הפעולה
בשפת Java — <code>public static int size (Queue&lt;Integer&gt; q)</code>	הפעולה מחזירה את מספר האיברים בתור q
בשפת C# — <code>public static int Size (Queue&lt;int&gt; q)</code>	בלי לשנות את התור.

א. שני תורים,  $q_1$  ו-  $q_2$ , יהיו "תורים זהים" אם מספר האיברים בשני התורים זהה, ובשני התורים מופיעים בדיוק אותם ערכים ובאותו הסדר.

דוגמה לשני תורים זהים:

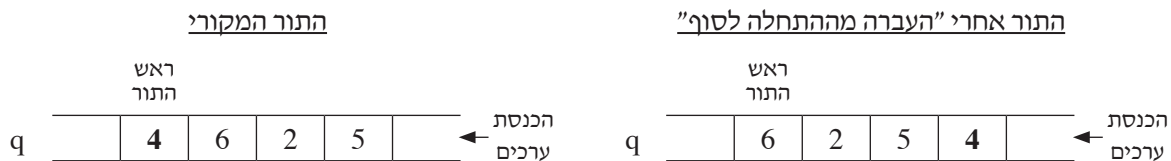


כתוב פעולה חיצונית ששמה `isIdentical` בשפת Java או `IsIdentical` בשפת C# המקבלת שני תורים מטיפוס שלם  $q_1$  ו-  $q_2$ , ומחזירה `true` אם התורים זהים, אחרת היא מחזירה `false`.

הערה: עם סיום הפעולה, חובה לשמור על מבנה התורים המקורי שהתקבל.

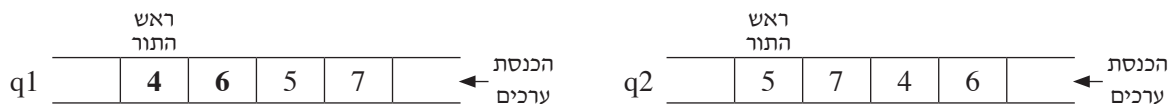
ב. "העברה מההתחלה לסוף" היא העברת מספר מראש התור לסופו.

דוגמה:



כתוב פעולה חיצונית ששמה `isSimilar` בשפת Java או `IsSimilar` בשפת C# המקבלת שני תורים מטיפוס שלם,  $q_1$  ו-  $q_2$ . הפעולה מחזירה `true` אם התורים  $q_1$  ו-  $q_2$  זהים — בין שהם זהים כמו שהם ובין שהם יהיו זהים לאחר שנבצע ב-  $q_1$  "העברה מההתחלה לסוף", פעם אחת או יותר. אחרת הפעולה מחזירה `false`.

דוגמה: הפעולה תחזיר `true` בעבור שני התורים  $q_1$  ו-  $q_2$  שלפניך:



זאת מכיוון שלאחר שבתור  $q_1$  נבצע פעמיים "העברה מההתחלה לסוף", הוא יהיה זהה לתור  $q_2$ , וייראה כך:



הערות: — חובה להשתמש בפעולה שכתבת בסעיף א.

— אין צורך לשמור על מבנה התורים המקורי שהתקבל.

## פרק שלישי

בפרק זה שאלות בארבעה מסלולים:

מערכות מחשב ואסמבלי, עמודים 18-19.

מבוא לחקר ביצועים, עמודים 20-22.

מודלים חישוביים, עמוד 23.

תכנות מונחה עצמים בשפת Java, עמודים 24-27; תכנות מונחה עצמים בשפת C#, עמודים 28-31.

אם בחרת לענות על שאלות מן הפרק הזה בחר בשאלות ממסלול אחד בלבד.

## מערכות מחשב ואסמבלי

8. בשאלה זו שני סעיפים, א-ב, שאין ביניהם קשר. עליך לענות על שניהם.

א. לפניך קטע תוכנית באסמבלי.

```
MOV AL, 6
MOV BX, 5CH
MOV DL, [BX]
DEC BX
A1: MOV CL, [BX]
INC CL
MOV [BX+1], CL
DEC BX
DEC AL
CMP AL, 0
JA A1
INC BX
INC DL
MOV [BX], DL
```

נתונה מפת תאי הזיכרון מ-56H עד 5CH לפני ביצוע קטע התוכנית:

address	56H	57H	58H	59H	5AH	5BH	5CH
value	59H	E2H	C5H	71H	ABH	3FH	8DH

(1) עקוב בעזרת טבלת מעקב אחר קטע התוכנית. בטבלת המעקב יש לכלול עמודה לכל אחד מן האוגרים

DL, CL, AL, BX. נוסף על כך, יש לכתוב מפת זיכרון מתאימה.

(2) הסבר מה מבצע קטע התוכנית.

ב. במקטע הנתונים הוגדר משתנה REZ בגודל מילה:

REZ DW ?

נתון שבמחסנית (ב- SP וב- SP+2) מאוחסנים שני מספרים מכוונים (signed) בגודל מילה.

כתוב קטע תוכנית המאחסן במשתנה REZ את המספר שהערך המוחלט שלו הוא הגדול מבין שניהם.

הערות: — הנח ששני המספרים שונים מ-8000H.

— הנח שהערך המוחלט של שני המספרים אינו זהה.

9. במקטע הנתונים הוגדרו הנתונים שלפניך:

ARR DB 10 DUP (?)

VAL DB ?

IND DB ?

כתוב קטע תוכנית שבו האיברים במערך — ARR החל מן האינדקס IND ועד סוף המערך מוזזים ימינה כך שהאיבר הימני ביותר יוצא מן המערך, והערך VAL מוכנס באינדקס IND. הנח שהמערך מלא במספרים, ש- VAL מאותחל וש-  $0 \leq \text{IND} \leq 9$ .

לדוגמה: בעבור  $\text{VAL} = 15$ ,  $\text{IND} = 5$  והמערך — ARR הזה:

index	0	1	2	3	4	5	6	7	8	9
value	-3	11	0	1	8	-9	5	6	33	73

יתקבל לאחר ביצוע קטע התוכנית המערך — ARR הזה:

index	0	1	2	3	4	5	6	7	8	9
value	-3	11	0	1	8	15	-9	5	6	33

**מבוא לחקר ביצועים**

10. א. (1) בטבלה שלפניך נתונה בעיית תובלה.

מקורות	יעדים				היצע
	1	2	3	4	
1	10	7	9	3	200
2	15	12	4	5	300
3	17	13	6	7	500
<b>ביקוש</b>	100	150	250	80	

העתק את הטבלה למחברתך והצע פתרון בסיסי אפשרי לפי שיטת הפינה הצפונית-מערבית.

(2) בטבלה שלפניך נתון פתרון בסיסי אפשרי לבעיית תובלה, ונתון הערך של  $u_1$ .

מקורות	יעדים				היצע	$u_i$
	1	2	3	4		
1	10 100	7 100	9	3	200	0
2	15	12 50	4 250	5	300	
3	17	13	6 20	7 180	200	
<b>ביקוש</b>	100	150	270	180		
<b><math>v_j</math></b>						

I. העתק את הטבלה למחברתך והשלם בה את הערכים  $u_2, u_3, v_1, v_2, v_3, v_4$ .

II. הוכח שהפתרון הנתון בטבלה אינו אופטימלי.

III. שנה את המחיר  $(c_{ij})$  באחד המשתנים הלא-בסיסיים כך שהפתרון הנתון בטבלה לבעיית התובלה

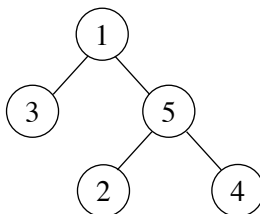
יהיה אופטימלי.

(שים לב: המשך השאלה בעמוד הבא.)

ב. גרף  $G = (V, E)$  לא מכיוון, קשיר ופשוט, ייקרא גרף "מתחלק" אם מתקיימים בו התנאים האלה:

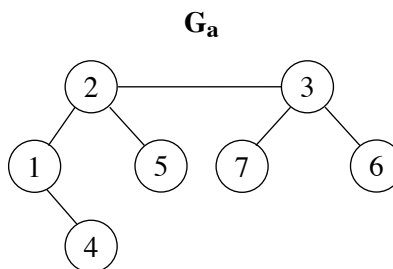
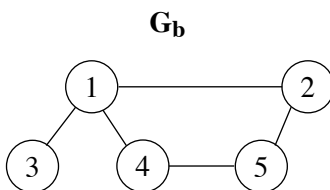
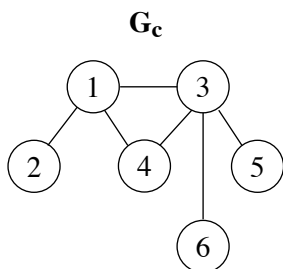
- יש בו שני צמתים לפחות.
- אפשר לחלק את הצמתים לשתי קבוצות, קבוצה א' וקבוצה ב', באופן שבו לא יהיו שני צמתים המחוברים בקשת באותה הקבוצה.

דוגמה:



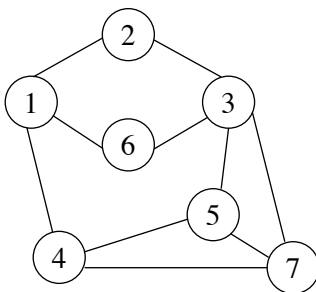
גרף זה הוא גרף "מתחלק" כי אפשר לחלק את הצמתים באופן שבו קבוצה א' תכיל את הצמתים  $\{3, 5\}$  וקבוצה ב' תכיל את הצמתים  $\{1, 2, 4\}$ .

(1) לפניך שלושה גרפים:  $G_a$ ,  $G_b$ ,  $G_c$ .



ציין איזה מן הגרפים הוא גרף "מתחלק" ואיזה אינו. בעבור כל גרף "מתחלק" הראה את שתי קבוצות הצמתים שבו.

(2) לפניך גרף שאינו "מתחלק". מהו מספר הקשתות המינימלי שיש להסיר כדי שהגרף ייחשב "מתחלק"? כתוב אילו קשתות יש להסיר והצג את שתי הקבוצות.



11. בשאלה זו שני סעיפים, א-ב, שאין ביניהם קשר. עליך לענות על שניהם.

א. לפניך מטריצת סמיכות המתארת גרף מכוון  $G = (V, E)$ .

	a	b	c	d	e	f
a	0	1	0	0	0	0
b	1	0	1	1	0	0
c	1	0	0	1	1	0
d	1	0	0	0	1	1
e	0	0	0	0	0	1
f	0	0	0	0	0	0

(1) סרטט את הגרף  $G$  בצורה גרפית.

(2) מצא את רכיבי הקשירות החזקה (רק"ח) בגרף.

(3) מהו מספר הקשתות המינימלי שיש להוסיף לגרף  $G$  כדי שיווצר רק"ח אחד בלבד? ציין את הקשתות שיש להוסיף.

ב. לפניך מטריצת סמיכות המתארת משקל מסלול בין צמתים בגרף מכוון ומשוקלל  $G = (V, E)$ .

	a	b	c	d	e
a	0	1	3	$\infty$	$\infty$
b	2	0	1	3	12
c	3	$\infty$	0	5	$\infty$
d	4	$\infty$	$\infty$	0	7
e	1	2	$\infty$	8	0

נתונות שתי נקודות התחלה אפשריות: a או c. עליך להגיע לצומת e מאחת משתי הנקודות האלה.

(1) מהי נקודת ההתחלה שממנה משקל המסלול לצומת e יהיה מינימלי?

עליך להפעיל את האלגוריתם של דייקסטרה ולהראות מעקב בכל איטרציה.

המעקב יכלול בכל איטרציה את קבוצת הצמתים הקבועים (P) ואת קבוצת הצמתים הזמניים (T).

(2) אם נשנה את המשקל של הקשת מצומת a לצומת c למשקל  $k$ ,  $k \geq 0$ , האם ישנה הפתרון? הסבר.

**מודלים חישוביים**

12.  $L$  היא שפת כל המילים מעל הא"ב  $\{a,b,c\}$  המקיימות את כל הדרישות שלפניך:

- האות שלפני האחרונה במילה היא  $a$ .
  - המילה מכילה לפחות פעמיים את הרצף  $bc$ .
  - מספר המופעים של האות  $b$  במילה הוא זוגי.
  - המילה אינה מכילה את הרצף  $bb$ .
- א. (1) הבא דוגמה אחת למילה השייכת לשפה  $L$ .
- (2) הבא דוגמה אחת למילה שאינה שייכת לשפה  $L$ .
- ב. הוכח שהשפה  $L$  רגולרית.
- הערה: אפשר להיעזר בתכונות סגירות.

13. נתונות השפות  $L_1$  ו-  $L_2$ :

$$L_1 = \{ a^n b^m c^k \mid n, k \geq 0, m = 2k \}$$

$$L_2 = \{ a^n b^m c^k \mid k, m, n \text{ גדולים מ-} 0 \text{ וצריכים להיות או כולם זוגיים או כולם אי-זוגיים} \}$$

- א. האם השפה  $L_1$  רגולרית? אם כן, בנה אוטומט סופי דטרמיניסטי מלא שמקבל את השפה, ואם לא, בנה אוטומט מחסנית שמקבל את השפה.
- ב. האם השפה  $L_2$  רגולרית? אם כן, בנה אוטומט סופי דטרמיניסטי מלא שמקבל את השפה, ואם לא, בנה אוטומט מחסנית שמקבל את השפה.

### תכנות מונחה עצמים בשפת Java

שאלות 14-15 מיועדות לכותבים בשפת Java.

14. כדי לנהל את האירועים שלה בנתה חברת "המחר" את המחלקה **Date** – תאריך, שלה שלוש תכונות מטיפוס שלם:

יום – day, חודש – month, שנה – year.

נוסף על כך, לצורך ניהול האירועים נבנתה המחלקה **Event** – אירוע.

לחברה יש שלושה סוגי אירועים: פגישות, שיחות טלפון ומשימות. כדי לשמור את המידע שבכל סוג אירוע, בנתה החברה

את המחלקות האלה: פגישה – **Meeting**, שיחת טלפון – **PhoneCall**, משימה – **Task**.

התכונות של המחלקות הן בהתאם לדרישות המידע שלפניך:

המידע ששומרים על פגישה – **Meeting**:

date – מטיפוס **Date**

hour – שעת ההתחלה, מטיפוס שלם

arrNames – שמות המשתתפים בפגישה, במערך מטיפוס מחרוזות (גודל המערך הוא לפי מספר המשתתפים)

duration – משך הפגישה בדקות, מטיפוס שלם

location – מקום הפגישה, מטיפוס מחרוזות

המידע ששומרים על שיחת טלפון – **PhoneCall**:

date – מטיפוס **Date**

hour – שעת ההתחלה, מטיפוס שלם

phoneNumber – מספר הטלפון שמחייגים, מטיפוס מחרוזות

name – שם האדם שאליו מתקשרים, מטיפוס מחרוזות

המידע ששומרים על משימה – **Task**:

date – מטיפוס **Date**

hour – שעת ההתחלה, מטיפוס שלם

title – כותרת המשימה, מטיפוס מחרוזות

המידע ששומרים על אירוע – **Event**:

אותן דרישות המידע המפורטות בסוגי האירועים שלעיל ושצריכות להישמר במחלקה **Event** בהתאם לעקרונות

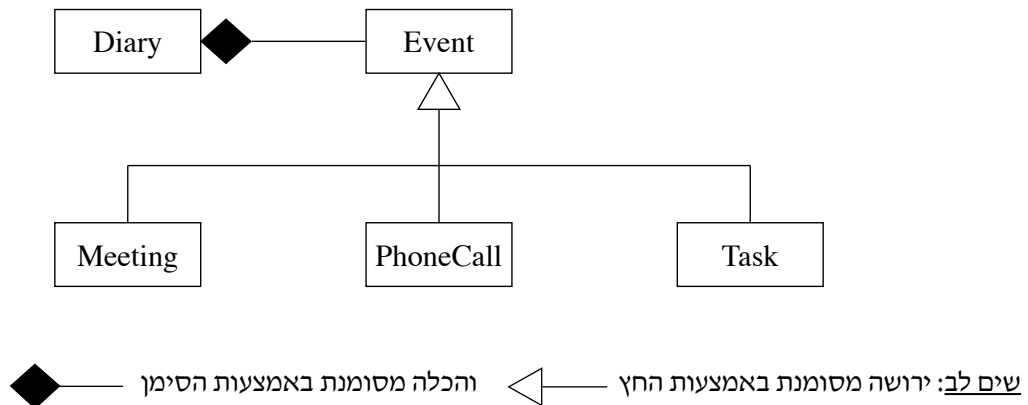
תכנות מונחה עצמים.

נוסף על המחלקות האמורות, נבנתה המחלקה **Diary** – יומן, השומרת עד 1,000 אירועים – **Event**,

במערך – arr בגודל 1,000 מטיפוס **Event**.



לפניך תרשים חלקי של המחלקות:



על הסעיפים שלפניך יש לענות לפי עקרונות של תכנות מונחה עצמים.

א. העתק את התרשים למחברתך והוסף את המחלקה **Date** לתרשים. כתוב את כותרת המחלקה אירוע — **Event** ואת התכונות שלה.

הנח שבשאר המחלקות כתובות התכונות המתאימות, ושכל המחלקות יש פעולות בונות ופעולות `get` ו-`set`.

ב. כתוב במחלקה **Diary** פעולה ששמה `allCalls` המקבלת עצם — `date` מטיפוס **Date**, ומחזירה מערך מטיפוס **PhoneCall** בגודל 100, המכיל את כל שיחות הטלפון בתאריך — `date`.

שים לב:

— הנח שקיימת הפעולה שלפניך במחלקה **Date**:

`public boolean same (Date other)`

הפעולה מחזירה `true` אם התאריך הנוכחי זהה ל-`other`, אחרת היא מחזירה `false`. אפשר להשתמש בפעולה בלי לממש אותה.

— הנח שיש בדיוק 100 שיחות טלפון בתאריך — `date`.

— הנח כי בעת הקריאה לפעולה זו, מערך האירועים — `arr` במחלקה **Diary** מלא בעצמים (ללא `null`).

ג. לפניך כותרת הפעולה הפנימית — `match` במחלקה **Event**:

`public boolean match (String name)`

זימון הפעולה יחזיר `true` אם האירוע (**Event**) הוא פגישה (**Meeting**) שבה משתתף אדם ששמו זהה ל-`name` או אם האירוע הוא שיחת טלפון (**PhoneCall**) עם אדם ששמו זהה ל-`name`, אחרת יחזיר `false`.

ממש את הפעולה, והוסף פעולות למחלקות המתאימות כדי לבצע את הנדרש.

ציין בעבור כל פעולה באיזו מחלקה היא מוספת.

הערה: בסעיף זה, אין להשתמש בפעולה `instanceof` ובפעולות של המחלקה `Object`.

15. לפניך המחלקות A, B, Tester :

```
public class A {
    public A() { System.out.println("ctor A"); }
    public void foo(int x) {
        System.out.println("A foo int " + x); }
    public void foo(double y) {
        System.out.println("A foo double " + y); }
    public void bar(int x) {
        System.out.println("A bar " + x);
        foo(x);
    }
}
```

```
public class B extends A {
    public B() { System.out.println("ctor B"); }
    public void foo(int x) {
        System.out.println("B foo int " + x); }
    public void bar() {
        System.out.println("B bar");
        foo(2);
    }
    public void another(int x) {
        System.out.println("B another " + x);
        super.foo(x);
        foo(2.0 * x);
    }
}
```

```
public class Tester {
    public static void main (String [] args)
    {
    }
}
```

(שים לב: המשך השאלה בעמוד הבא.)

לפניך 15 קטעי קוד. בחר עשרה מהם.

הצב כל אחד מקטעי הקוד שבחרת בפעולה main במחלקה Tester, כתוב את מספר הקטע במחברתך וציין אם הקוד תקין או אינו תקין. אם הקוד תקין – כתוב את הפלט, ואם הוא אינו תקין – הסבר את השגיאה. קטעי הקוד אינם קשורים זה לזה.

<b>1.</b> A a = new A(); a.foo(2);	<b>2.</b> A a = new A(); ((B)a).foo(3);	<b>3.</b> A a = new A(); B b = a; b.foo(2);
<b>4.</b> A x = new B(); x.foo(2);	<b>5.</b> A x = new B(); x.bar();	<b>6.</b> A a = new A(); a.bar(3);
<b>7.</b> A a = new A(); a.bar();	<b>8.</b> B b = new B(); b.bar();	<b>9.</b> B b = new B(); b.bar(3);
<b>10.</b> B b = new B(); b.foo(2); b.foo(2.0);	<b>11.</b> A a = new A(); a.foo(2); a.foo(2.0);	<b>12.</b> B b = new A(); b.another(2);
<b>13.</b> A a = new A(); a.another(2);	<b>14.</b> B x = new B(); x.another(2);	<b>15.</b> A x = new B(); x.another(2);

תכנות מונחה עצמים בשפת C#

שאלות 16-17 מיועדות לכותבים בשפת C#.

16. כדי לנהל את האירועים שלה בנתה חברת "המחר" את המחלקה **Date** – תאריך, שלה שלוש תכונות מטיפוס שלם:

יום – day, חודש – month, שנה – year.

נוסף על כך, לצורך ניהול האירועים נבנתה המחלקה **Event** – אירוע.

לחברה יש שלושה סוגי אירועים: פגישות, שיחות טלפון ומשימות. כדי לשמור את המידע שבכל סוג אירוע, בנתה החברה

את המחלקות האלה: פגישה – **Meeting**, שיחת טלפון – **PhoneCall**, משימה – **Task**.

התכונות של המחלקות הן בהתאם לדרישות המידע שלפניך:

המידע ששומרים על פגישה – **Meeting**:

date – מטיפוס **Date**

hour – שעת ההתחלה, מטיפוס שלם

arrNames – שמות המשתתפים בפגישה, במערך מטיפוס מחרוזת (גודל המערך הוא לפי מספר המשתתפים)

duration – משך הפגישה בדקות, מטיפוס שלם

location – מקום הפגישה, מטיפוס מחרוזת

המידע ששומרים על שיחת טלפון – **PhoneCall**:

date – מטיפוס **Date**

hour – שעת ההתחלה, מטיפוס שלם

phoneNumber – מספר הטלפון שמחייגים, מטיפוס מחרוזת

name – שם האדם שאליו מתקשרים, מטיפוס מחרוזת

המידע ששומרים על משימה – **Task**:

date – מטיפוס **Date**

hour – שעת ההתחלה, מטיפוס שלם

title – כותרת המשימה, מטיפוס מחרוזת

המידע ששומרים על אירוע – **Event**:

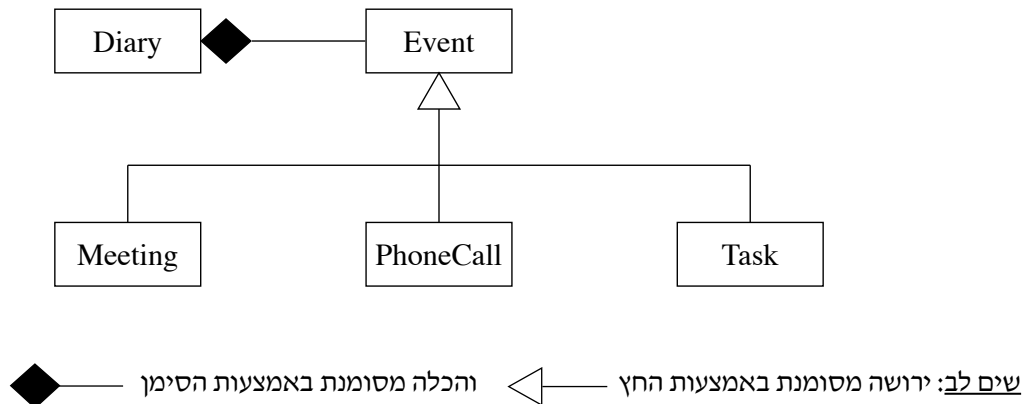
אותן דרישות המידע המפורטות בסוגי האירועים שלעיל ושצריכות להישמר במחלקה **Event** בהתאם לעקרונות

תכנות מונחה עצמים.

נוסף על המחלקות האמורות, נבנתה המחלקה **Diary** – יומן, השומרת עד 1,000 אירועים – **Event**,

במערך – arr בגודל 1,000 מטיפוס **Event**.

לפניך תרשים חלקי של המחלקות:



על הסעיפים שלפניך יש לענות לפי עקרונות של תכנות מונחה עצמים.

א. העתק את התרשים למחברתך והוסף את המחלקה **Date** לתרשים. כתוב את כותרת המחלקה אירוע — **Event** ואת התכונות שלה.

הנח שבשאר המחלקות כתובות התכונות המתאימות, ושכל המחלקות יש פעולות בונות ופעולות Get ו- Set .

ב. כתוב במחלקה **Diary** פעולה ששמה AllCalls המקבלת עצם — date מטיפוס **Date** , ומחזירה מערך מטיפוס **PhoneCall** בגודל 100 , המכיל את כל שיחות הטלפון בתאריך — date .

שים לב:

— הנח שקיימת הפעולה שלפניך במחלקה **Date** :

public bool Same (Date other)

הפעולה מחזירה true אם התאריך הנוכחי זהה ל- other , אחרת היא מחזירה false . אפשר להשתמש בפעולה בלי לממש אותה.

— הנח שיש בדיוק 100 שיחות טלפון בתאריך — date .

— הנח כי בעת הקריאה לפעולה זו, מערך האירועים — arr במחלקה **Diary** מלא בעצמים (ללא null).

ג. לפניך כותרת הפעולה הפנימית — Match במחלקה **Event** :

public virtual bool Match (string name)

זימון הפעולה יחזיר true אם האירוע (**Event**) הוא פגישה (**Meeting**) שבה משתתף אדם ששמו זהה ל- name או אם האירוע הוא שיחת טלפון (**PhoneCall**) עם אדם ששמו זהה ל- name , אחרת יוחזר false .

ממש את הפעולה, והוסף פעולות למחלקות המתאימות כדי לבצע את הנדרש. ציין בעבור כל פעולה באיזו מחלקה היא מוספת.

הערה: בסעיף זה, אין להשתמש בפעולות is ו- as ובפעולות של המחלקה Object .

17. לפניך המחלקות A, B, Tester :

```
public class A {
    public A() { Console.WriteLine("ctor A"); }
    public virtual void Foo(int x) {
        Console.WriteLine("A Foo int " + x); }
    public void Foo(double y) {
        Console.WriteLine("A Foo double " + y); }
    public void Bar(int x) {
        Console.WriteLine("A Bar " + x);
        Foo(x);
    }
}
```

```
public class B : A {
    public B() { Console.WriteLine("ctor B"); }
    public override void Foo(int x) {
        Console.WriteLine("B Foo int " + x); }
    public void Bar() {
        Console.WriteLine("B Bar");
        Foo(2);
    }
    public void Another(int x) {
        Console.WriteLine("B Another " + x);
        base.Foo(x);
        Foo(2.0 * x);
    }
}
```

```
public class Tester {
    public static void Main (string [] args)
    {
    }
}
```

(שים לב: המשך השאלה בעמוד הבא.)

לפניך 15 קטעי קוד. בחר עשרה מהם.

הצב כל אחד מקטעי הקוד שבחרת בפעולה Main במחלקה Tester, כתוב את מספר הקטע במחברתך וציין אם הקוד תקין או אינו תקין. אם הקוד תקין – כתוב את הפלט, ואם הוא אינו תקין – הסבר את השגיאה. קטעי הקוד אינם קשורים זה לזה.

<b>1.</b> A a = new A(); a.Foo(2);	<b>2.</b> A a = new A(); ((B)a).Foo(3);	<b>3.</b> A a = new A(); B b = a; b.Foo(2);
<b>4.</b> A x = new B(); x.Foo(2);	<b>5.</b> A x = new B(); x.Bar();	<b>6.</b> A a = new A(); a.Bar(3);
<b>7.</b> A a = new A(); a.Bar();	<b>8.</b> B b = new B(); b.Bar();	<b>9.</b> B b = new B(); b.Bar(3);
<b>10.</b> B b = new B(); b.Foo(2); b.Foo(2.0);	<b>11.</b> A a = new A(); a.Foo(2); a.Foo(2.0);	<b>12.</b> B b = new A(); b.Another(2);
<b>13.</b> A a = new A(); a.Another(2);	<b>14.</b> B x = new B(); x.Another(2);	<b>15.</b> A x = new B(); x.Another(2);

### בהצלחה!