

## מדעי המחשב ב'

2 יחידות לימוד (השלמה ל-5 יח"ל)

### הוראות לנבחן

א. משך הבחינה: שלוש שעות.

ב. מבנה השאלון ומפתח ההערכה: בשאלון זה שני פרקים.

פרק ראשון — בפרק זה ארבע שאלות, — (25x2) — 50 נקודות  
ומהן יש לענות על שתיים.

פרק שני — בפרק זה שאלות בארבעה מסלולים שונים. — (25x2) — 50 נקודות  
ענה על שאלות רק במסלול שלמדת,  
לפי ההוראות בקבוצת השאלות במסלול זה.

סה"כ — 100 נקודות

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

ד. הוראות מיוחדות:

(1) את כל התכניות שאתה נדרש לכתוב בשפת מחשב בפרק הראשון  
כתוב בשפה אחת בלבד — Java או C#.

(2) רשום על הכריכה החיצונית של המחברת באיזו שפה אתה כותב — Java או C#.

(3) 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 בטיטה (ראשי פרקים, חישובים וכדומה).  
רשום "טייטה" בראש כל עמוד טייטה. רישום טייטות כלשהן 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 שני פרקים: פרק ראשון ופרק שני.  
עליך לענות על שאלות משני הפרקים, לפי ההוראות בכל פרק.

### פרק ראשון (50 נקודות)

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

ענה על שתיים מן השאלות 1-4 (לכל שאלה — 25 נקודות).

1. לפניך הגדרה של חמש פעולות הפועלות על מבנה נתונים כלשהו.  
**שים לב:** שמות הפעולות שלפניך אינם כתובים ב-Java או ב-C#.  $\text{insert}(x)$  — פעולה המכניסה למבנה איבר שערכו  $x$  מטיפוס שלם.  
 $\text{showMin}()$  — פעולה המחזירה את הערך הנמוך ביותר במבנה, בלי לשנות את המבנה.  
 $\text{getMax}()$  — פעולה המחזירה את האיבר שערכו הוא הגדול ביותר במבנה, ומוציאה אותו מן המבנה. אם יש יותר מאיבר אחד כזה, הפעולה תחזיר ותוציא את זה שמופיע ראשון.  
 $\text{exists}(x)$  — פעולה בוליאנית המחזירה  $\text{true}$  אם האיבר שערכו  $x$  קיים במבנה. אחרת — הפעולה מחזירה  $\text{false}$ .  
 $\text{div7}()$  — פעולה בוליאנית המחזירה  $\text{true}$  אם קיים במבנה איבר שערכו מתחלק ב-7 בלי שארית. אחרת — הפעולה מחזירה  $\text{false}$ .  
נרצה להציע מבני נתונים העומדים בדרישות סיבוכיות שונות למימוש פעולות מתוך חמש הפעולות שהוגדרו.  
דוגמה: רוצים להציע מבנה נתונים שאפשר לבצע עליו את הפעולות  $\text{insert}$ ,  $\text{showMin}$  בסיבוכיות  $O(1)$ , ואת הפעולות  $\text{getMax}$ ,  $\text{exists}$  בסיבוכיות  $O(n)$ . לשם כך נגדיר את מבנה הנתונים ונסביר כיצד ימומשו הפעולות.  
**שים לב:** במבנה זה אין צורך להתייחס לפעולה  $\text{div7}$ .
- (שים לב: המשך השאלה בעמוד הבא.)

מבנה נתונים מתאים מורכב מ:

- רשימה מקושרת דו-כיוונית,  $lst$  מטיפוס שלם.
- מצביע על האיבר המינימלי שהוכנס לרשימה,  $min$ .

הפעולות יבוצעו כך:

הפעולה	הסבר כיצד תמומש	נימוק מדוע המימוש עומד בדרישות הסיבוכיות
<b>insert(x)</b>	<ul style="list-style-type: none"> <li>— הכנסת האיבר <math>x</math> לראש הרשימה.</li> <li>— אם האיבר קטן מן המינימום עד כה, עדכון המצביע על <math>min</math> כך שיצביע על האיבר החדש.</li> </ul>	<ul style="list-style-type: none"> <li>— הכנסת איבר לראש הרשימה — <math>O(1)</math>.</li> <li>— אם הוכנס איבר שקטן מן המינימום: עדכון המצביע לאיבר המינימלי — <math>O(1)</math>.</li> <li>— אם הוכנס איבר ראשון — הכנסה לראש הרשימה ב- <math>O(1)</math> ועדכון המצביע לאיבר הראשון שהוא גם המינימלי.</li> <li>— סך הכול — <math>O(1)</math>.</li> </ul>
<b>showMin()</b>	החזרת הערך של האיבר שעליו מצביע $min$ .	החזרת הערך — $O(1)$ .
<b>exists(x)</b>	מעבר על הרשימה $lst$ וחיפוש האיבר שערכו $x$ .	במקרה הגרוע — מעבר על כל הרשימה — $O(n)$ .
<b>getMax()</b>	מעבר על הרשימה $lst$ , חיפוש האיבר שערכו מקסימלי והוצאתו מן הרשימה.	<ul style="list-style-type: none"> <li>— חיפוש האיבר שערכו מקסימלי — <math>O(n)</math>.</li> <li>— הוצאתו מן הרשימה — <math>O(1)</math>.</li> </ul>

לפניך שני סעיפים א-ב. בעבור כל אחד מן הסעיפים עליך להציע מבנה נתונים מתאים העומד

בדרישות המפורטות בסעיף. המבנה יכול להיות מורכב משילוב של כמה מבנים וטיפוסים

שלמדת.

לכל אחת מן הפעולות הסבר כיצד תממש אותה, ונמק מדוע המימוש עומד בדרישות (כפי שהוצג

בטבלה שבדוגמה). אין צורך לממש את הפעולות.

א. ביצוע הפעולות  $insert$ ,  $exists$  בסיבוכיות  $O(n)$ , וביצוע הפעולות  $showMin$ ,  $getMax$  בסיבוכיות  $O(1)$ .

ב. ביצוע הפעולות  $insert$ ,  $getMax$  בסיבוכיות  $O(n)$ , וביצוע הפעולה  $div7$  בסיבוכיות  $O(1)$ .

2.

לפניך תכנית הכתובה ב-Java וב-C#.

הפעולה what ב-Java ו-What ב-C# מקבלת מערך חד-ממדי המכיל מספרים שלמים,

ומספר שלם  $k$ ,  $k > 0$ , כל המספרים במערך הם בין 0 ל- $k$  (כולל).

### Java

```
public class Program
{
    public static int[] what(int[] arr, int k)
    {
        int n = arr.length;
        int[] b = new int [n];
        int[] c = new int [k+1];

        for (int i = 0; i < k+1; i++) c[i] = 0;
        for (int j = 0; j < n; j++) c[arr[j]] = c[arr[j]] + 1;           /**

        for (int i = 1; i < k+1 ; i++) c[i] = c[i] + c[i-1];           /**

        for (int j = n-1 ; j >= 0; j--)                               /**
        {
            b[c[arr [j]] - 1] = arr[j];
            c[arr[j]] = c[arr[j]] - 1;
        }
        return b;
    }

    public static void main(String[] args)
    {
        int[] arr = new int[] {5,0,2,1,3,0,5};
        arr = what(arr, 5);
        for (int i = 0; i < arr.length; i++) System.out.println(arr[i]);
    }
}
```

C#

```

public class Program
{
    public static int[] What(int[] arr, int k)
    {
        int n = arr.Length;
        int[] b = new int [n];
        int[] c = new int [k+1];

        for (int i = 0; i < k+1; i++) c[i] = 0;
        for (int j = 0; j < n; j++) c[arr[j]] = c[arr[j]] + 1;           /**

        for (int i = 1; i < k+1 ; i++) c[i] = c[i] + c[i-1];           /***

        for (int j = n-1 ; j >= 0; j--)                                /****
        {
            b[c[arr[j]] - 1] = arr[j];
            c[arr[j]] = c[arr[j]] - 1;
        }
        return b;
    }

    public static void Main(string[] args)
    {
        int[] arr = new int[] {5,0,2,1,3,0,5};
        arr = What(arr, 5);
        for (int i = 0; i < arr.Length; i++) Console.WriteLine(arr[i]);
    }
}

```

- א. (1) סרטט את המערך c לאחר ביצוע הלולאה המסומנת /\*\*.
- (2) סרטט את המערך c לאחר ביצוע הלולאה המסומנת /\*\*\*.
- (3) עקוב אחר ביצוע הלולאה המסומנת /\*\*\*\*.

במעקב עליך להראות את: j , arr[j] , c[arr[j]] - 1 , המערך b והמערך c.

ב. מה מבצעת הפעולה what ב-Java או What ב-C# ?

ג. מהי סיבוכיות זמן הריצה של הפעולה what ב-Java או What ב-C# ? נמק את תשובתך.  
/המשך בעמוד 6/

3. א. ממש פעולה חיצונית exist ב-Java או Exist ב-C#.

הפעולה תקבל עץ בינרי t מטיפוס שלם ומספר שלם x. הפעולה תחזיר true אם יש בעץ צומת שערכו x, אחרת – הפעולה תחזיר false. אם העץ ריק – הפעולה תחזיר false.

ב. לפניך הפעולה ב-Java check(t1, t2) וב-C# Check(t1, t2).

הפעולה מקבלת שני עצים בינריים לא ריקים מטיפוס שלם, t1 ו-t2, ומחזירה רשימה המכילה את כל המספרים הנמצאים בעץ t1 ואינם נמצאים בעץ t2. הפעולה מזמנת פעולה נוספת המקבלת שלושה פרמטרים.

Java

```
public static Node<Integer> check(BinNode<Integer> t1, BinNode<Integer> t2)
{
    Node<Integer> first = new Node<Integer> (- 1);
    first = check(t1 , t2 , first);
    return first.getNext();
}
```

C#

```
public static Node<int> Check(BinNode<int> t1 , BinNode<int> t2)
{
    Node<int> first = new Node<int> (- 1);
    first = Check(t1 , t2 , first);
    return first.GetNext();
}
```

ממש את הפעולה:

ב-Java:

```
public static Node<Integer> check( BinNode<Integer> t1,
                                   BinNode<Integer> t2 , Node<Integer> list)
```

או ב-C#:

```
public static Node<int> Check(BinNode<int> t1 , BinNode<int> t2 , Node<int> list)
```

אתה יכול להשתמש בפעולה שמימשת בסעיף א.

ג. מה היא סיבוכיות זמן הריצה של הפעולה שמימשת בסעיף ב ? נמק את תשובתך.

/המשך בעמוד 7/

4.

תמונה הנראית על מסך המחשב מורכבת מנקודות צבעוניות — פיקסלים. כל פיקסל מורכב משלושה רכיבי צבע: אדום, ירוק, כחול (Red Green Blue — RGB). כל רכיב צבע מיוצג על ידי מספר שלם בין 0 ל-255, והשילוב של שלושת הרכיבים קובע את צבע הפיקסל. לדוגמה: צבע לבן מיוצג על ידי המספרים (255, 255, 255), צבע אדום מיוצג על ידי המספרים (255, 0, 0), צבע כחול — (0, 0, 255), צבע שחור — (0, 0, 0).

א. כתוב ב־Java או ב־C# את הכותרת ואת התכונות של המחלקה **Pixel** שמייצגת פיקסל. לפיכך תרשים חלקי של ממשק המחלקה **Pixel** הכתוב ב־Java וב־C#.

פעולה בונה היוצרת את הצבע השחור.	public Pixel()
פעולה בונה המקבלת 3 מספרים שלמים שהם הערכים של שלושת רכיבי הצבע של הפיקסל. אם אחד (לפחות) מן הפרמטרים לא נמצא בתחום הערכים המותר (0-255 [כולל]) — ייבנה עצם שצבעו שחור.	public Pixel(int red , int green , int blue)
הפעולה מקבלת עצם מטיפוס <b>Pixel</b> , ומשנה את ערכי הצבעים של העצם הנוכחי לצבע המתקבל מערבוב של שני הצבעים (הצבע של העצם הנוכחי והצבע של העצם שהתקבל כפרמטר). ערבוב הצבעים המתקבל הוא ממוצע של שני הרכיבים.	Java: public void mix(Pixel pxl) C#: public void Mix(Pixel pxl)

**שים לב:** כאשר מתקבל ממוצע שהוא שבר (למשל 113.5) נלקח רק החלק השלם של הערך החדש (במקרה זה — 113). לדוגמה: אם הצבע של העצם הנוכחי הוא אדום (255, 0, 0) והצבע של העצם שהתקבל כפרמטר הוא ירוק (0, 255, 0), אז הצבע המתקבל מן הערבוב הוא צהוב (127, 127, 0), והוא יהיה מעתה הצבע של העצם הנוכחי.

ב. ממש ב־Java או ב־C# את הפעולות המופיעות בממשק המחלקה **Pixel**.

ג. (1) הוסף מחלקה בשם **Pic50** שמייצגת תמונה בגודל  $50 \times 50$ .

למחלקה התכונה: pic — מערך דו־ממדי בגודל  $50 \times 50$  מטיפוס **Pixel**. כתוב ב־Java או ב־C# את כותרת המחלקה ואת התכונה שלה.

(2) כתוב במחלקה **Pic50** פעולה mix ב־Java או Mix ב־C#, שתקבל תמונה. הפעולה תערבב את הצבעים של התמונה שהתקבלה עם הצבעים של התמונה הנוכחית ותעדכן את התמונה הנוכחית. הערבוב ייעשה בין פיקסלים שהאינדקסים שלהם זהים.

## פרק שני (50 נקודות)

בפרק זה שאלות בארבעה מסלולים:

מערכות מחשב ואסמבלי, עמודים 8-13.

מבוא לחקר ביצועים, עמודים 14-21.

מודלים חישוביים, עמודים 22-24.

תכנות מונחה עצמים ב-Java, עמ' 25-32; תכנות מונחה עצמים ב-C#, עמודים 33-40.

ענה רק על שאלות במסלול שלמדת.

### מערכות מחשב ואסמבלי

אם למדת מסלול זה, ענה על שתיים מן השאלות 5-8 (לכל שאלה — 25 נקודות).

5. בשאלה זו שני סעיפים, א-ב. אין קשר בין הסעיפים. עליך לענות על שניהם.

א. לפניך קטע תכנית באסמבלי שבו ב-AL וב-AH יש מספרים שונים זה מזה לא מסומנים.

```
MOV AX, 7F80H
CMP AH, AL
JL SMALL
BIG: MOV BL, AH
JMP SOF
SMALL: MOV BL, AL
SOF:
```

(1) מה יהיה ערכו של BL בסיום ביצוע קטע התכנית?

(2) מה מבצע קטע התכנית?

(3) בקטע התכנית, במקום ההוראה JL SMALL נכתבה ההוראה JB SMALL.

האם בסיום הביצוע של קטע התכנית לאחר ההחלפה יהיה ערכו של BL שונה מערכו

בסיום ביצוע הקטע לפני החלפת ההוראה? הסבר את תשובתך.



ב. (אין קשר לסעיף א.)

לפניך שישה היגדים. לכל אחד מהם קבע אם הוא נכון או אינו נכון. אם ההיגד אינו נכון, הסבר מדוע.

$$10011110101110_2 > 27AE_{16} \quad (1)$$

$$27AE_{16} > 10159_{10} \quad (2)$$

(3) התפקידים של אוגרים BP ו- SP הם:

BP קובע באיזו כתובת תירשם מילה במחסנית, ומאיזו כתובת במחסנית תישלף מילה.

SP מאפשר לשגרה (פרוצדורה) לשלוף פרמטרים מן המחסנית, בלי לערב את BP.

(4) בסיום הקריאה והביצוע של כל הוראה, ערכו של IP גדל תמיד ב- 1.

(5) לפניך שני קטעי תכנית באסמבלי.

קטע 2	קטע 1
MOV DX , 0000H	MOV DX , 0000H
MOV AX , 0064H	MOV AX , 0064H
DIV AX	DIV AL

בסיום ביצוע שני הקטעים התוכן של AX יהיה 0001H.

(6) לפניך שני קטעי תכנית באסמבלי.

קטע 2	קטע 1
MOV AL , 5BH	MOV AL , 5BH
MOV CL , 9	MOV CL , 8
ROL AL , CL	ROL AL , CL

בסיום ביצוע קטע 1 ערכו של AL יהיה שווה לערך שהוצב בו בהוראה הראשונה,

וגם בסיום ביצוע קטע 2 ערכו של AL יהיה שווה לערך שהוצב בו בהוראה הראשונה.

6. א. כתוב באסמבלי שגרה (פרוצדורה) בשם CALC שתקבל כפרמטר מספר בגודל מילה. המספר קטן מ-  $9_{10}$  וגדול מ- 0. השגרה תחשב את העצרת של מספר זה, ותאחסן את התוצאה כמילה-כפולה המורכבת מצמד האוגרים DX:AX.

ב. במקטע הנתונים נכתב:

ARR DW 3, 10, 0, 8

RES DW 8 DUP (0)

- ARR הוא מערך של 4 מספרים שכל אחד מהם בגודל מילה;
- RES הוא מערך של 4 התוצאות שתחזיר השגרה CALC. כל תוצאה היא בגודל מילה-כפולה.

(1) כתוב תכנית שתבצע את המשימות האלה:

- התכנית תקרא את 4 המספרים מן המערך ARR.
- אם מספר קטן מ-  $9_{10}$  וגם גדול מ- 0 — התכנית תשלח אותו כפרמטר לשגרה CALC שכתבת בסעיף א. התכנית תרשום את התוצאה שתחזיר השגרה כמספר הקסדצימלי בגודל מילה-כפולה, לפי הסדר, במערך התוצאות RES.
- אם המספר אינו בתחום הנדרש — התכנית תכתוב לפי הסדר במערך התוצאות RES את המילה 0000H, ולאחריה — את המספר כמספר הקסדצימלי.

- (2) העתק למחברתך את הטבלה שלפניך, והשלם אותה כשכתובות הקיסט של תאי המערך RES מתחילות בהיסט 0010H.

הכתובות	הערך שתחזיר השגרה CALC
0010 - 11	
0012 - 13	
0014 - 15	
0016 - 17	

7. נתונים שני מערכים ARR1 ו-ARR2. שני המערכים הם בגודל שווה וכל אחד מן האיברים שלהם הוא בגודל מילה.
- בכל איבר, בכל אחד מן המערכים, מאוחסן מספר הקסדיצמלי שלם בן ארבע ספרות, בתחום 1000H - FFFFH.
- כמו כן נתון המשתנה K בגודל בית שבו מאוחסן גודל המערכים ARR1 ו-ARR2.
- כתוב באסמבלי קטע תכנית שיאחסן באוגר DX את מספר זוגות המספרים (האחד ממערך ARR1 והאחר ממערך ARR2) שמקיימים את התנאים האלה:
- (i) לשני המספרים בזוג יש אותו מציין (אינדקס) במערך.
- (ii) שני המספרים בזוג מורכבים מאותן ספרות אך בסדר הפוך.
- דוגמה: בעבור המערכים ARR1 ו-ARR2 בגודל 5 שלפניך, יאוחסן באוגר DX המספר 2.

ARR1	2025H	1061H	1492H	5777H	1948H
ARR2	1984H	1601H	2914H	9999H	8491H

8. א. במקטע נתונים הוגדרו הנתונים כך:

```
ARR DB 04H, 0FH, 07H
```

שים לב: המספרים הם הקסדצימליים.

לפניך תכנית באסמבלי.

```
BEGIN: MOV SI, 0
        MOV BP, 10H
        MOV CX, 2

NEXT1:  MOV DI, BP
        XOR AX, AX
        MOV AL, [SI]
        MOV [DI], AL
        MOV BX, AX

        MOV DL, 2
        DIV DL
        MOV DL, AL

CHECK:  MOV AX, BX
        DIV DL
        CMP AH, 0
        JNZ NEXT2

        INC DI
        MOV [DI], DL
NEXT2:  DEC DL
        JNZ CHECK

        INC SI
        ADD BP, 10H
        LOOP NEXT1
```

SOF:

(שים לב: סעיפי השאלה בעמוד הבא.)

מפת מקטע הנתונים:

0 1 2 3 4 5 6 7 8 9 A B C D E F

DS: 0000 04 0F

DS: 0010 \_ \_ \_

DS: \_ \_ \_ \_ \_

(1) עקוב בעזרת טבלת מעקב אחר ביצוע התכנית בעבור המעריך ARR הנתון,

וכתוב מה יהיה התוכן של:

SI , BP , CX , DI , AH , AL , BX , DL.

(2) העתק למחברתך את המפה של מקטע הנתונים, והשלם אותה בעבור כל הנתונים

המתקבלים במהלך ביצוע התכנית בעבור המעריך ARR הנתון.

(3) כתוב מה התכנית מבצעת.

ב. (1) מחקו מן התכנית את ההוראה: XOR AX, AX.

האם תבצע התכנית את הנדרש? הסבר את תשובתך.

(2) החליפו בתכנית את ההוראה XOR AX, AX בהוראה XOR AL, AL.

האם תבצע התכנית את הנדרש? הסבר את תשובתך.

(3) החליפו בתכנית את ההוראה XOR AX, AX בהוראה XOR AH, AH.

האם תבצע התכנית את הנדרש? הסבר את תשובתך.

### מבוא לחקר ביצועים

אם למדת מסלול זה, ענה על שתיים מן השאלות 9-12 (לכל שאלה – 25 נקודות).

9. נתונה בעיית תכנון לינארי:

$$\max \{z = 5x_1 - x_2\}$$

בכפוף לאילוצים האלה:

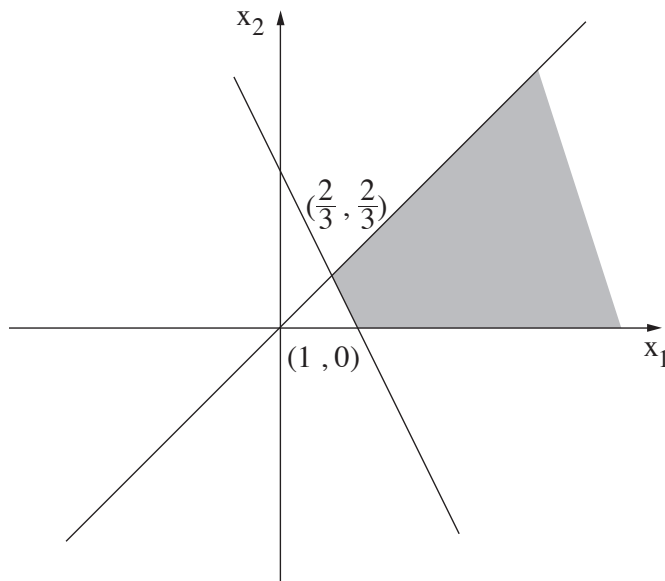
$$x_2 \leq x_1$$

$$2x_1 + x_2 \geq 2$$

$$x_1 \geq 0$$

$$x_2 \geq 0$$

לפניך סרטוט של תחום הפתרונות האפשריים של הבעיה הנתונה.



כל אחד מן הסעיפים א-ה שבעמוד הבא מתייחס לבעיית התכנון הלינארי הנתונה.

הסעיפים א-ה אינם קשורים זה לזה. ענה על כל הסעיפים.

בעבור הסעיפים א-ד נתונים ארבעה היגדים iv-i, ולכל סעיף רק היגד אחד נכון.

i הפתרון האופטימלי הוא יחיד.

ii יש אין-סוף פתרונות אופטימליים.

iii הפתרון האופטימלי לא חסום.

iv אין פתרון אופטימלי.

בכל אחד מן הסעיפים א-ד קבע איזה היגד הוא הנכון, העתק אותו למחברתך, ונמק את קביעתך.

- אם בחרת בהיגד i בסעיף כלשהו, עליך למצוא את הפתרון האופטימלי היחיד ואת הערך של פונקציית המטרה בפתרון זה.
  - אם בחרת בהיגד ii בסעיף כלשהו, עליך לרשום את הפתרון האופטימלי הכללי לבעיה ואת הערך של פונקציית המטרה בתחום הפתרונות האופטימליים.
- א. איזה היגד הוא הנכון בעבור בעיית תכנון לינארי הנתונה בתחילת השאלה? נמק את תשובתך.

ב. משנים רק את פונקציית המטרה של הבעיה הנתונה בתחילת השאלה

$$\text{ל- } \max \{z = -5x_1 + x_2\}.$$

איזה היגד הוא הנכון לאחר שינוי פונקציית המטרה? נמק את תשובתך.

ג. משנים רק את פונקציית המטרה של הבעיה הנתונה בתחילת השאלה

$$\text{ל- } \min \{z = -5x_1 + x_2\}.$$

איזה היגד הוא הנכון לאחר שינוי פונקציית המטרה? נמק את תשובתך.

ד. משנים רק את פונקציית המטרה של הבעיה הנתונה בתחילת השאלה

$$\text{ל- } \max \{z = -2x_1 + 2x_2\}.$$

איזה היגד הוא הנכון לאחר שינוי פונקציית המטרה? נמק את תשובתך.

ה. מוסיפים אילוץ נוסף לבעיה הנתונה בתחילת השאלה והוא:  $2x_1 - x_2 \leq 2$

ומשנים את פונקציית המטרה של הבעיה הנתונה בתחילת השאלה ל-  $\max \{z = ax_1 + x_2\}$

כאשר  $-1 < a < 2$ , והערך המקסימלי שמקבלת פונקציית המטרה הוא 4 בתחום

הפתרונות האפשריים.

חשב את הערך של  $a$ .

10. בשאלה זו שני סעיפים, א-ב. אין קשר בין הסעיפים. עליך לענות על שניהם.
- א.  $G = (V, E)$  הוא גרף מכוון המיוצג על ידי מטריצת הסמיכויות שלפניך:

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

- (1) סרטט את הגרף  $G$  המיוצג על ידי המטריצה.
- (2) מצא את רכיבי הקשירות החזקה (רק"חים) שבגרף הנתון. בעבור כל רק"ח שמצאת, רשום את קבוצת הקדקודים שלו.
- (3) מהו המספר המקסימלי של קשתות שאפשר להסיר מן הגרף הנתון כדי שיכיל אותו מספר רק"חים שמצאת בסעיף הקודם? מהי הקשת או מה הן הקשתות שאפשר להסיר?
- ב. (אין קשר לסעיף א.)
- $G = (V, E)$  הוא גרף לא מכוון המיוצג על ידי רשימת הסמיכויות שלפניך:

a	→	b	→	c	→	d	→	
b	→	a	→	c	→			
c	→	a	→	b	→			
d	→	a	→	f	→	e	→	
e	→	d	→	f	→			
f	→	e	→	d	→			

- (1) סרטט את הגרף  $G$  המיוצג על ידי רשימת הסמיכויות.
- (2) האם הגרף הנתון  $G$  הוא גרף קשיר? נמק.
- (3) הפעל אלגוריתם סריקה לעומק (DFS) על הגרף הנתון החל בקדקוד a. סרטט במחברתך רק את העץ הפורש (DFS) שמתקבל. התבסס על הייצוג הנתון על ידי רשימת הסמיכויות.
- (4) הפעל אלגוריתם סריקה לרוחב (BFS) על הגרף הנתון החל בקדקוד a. סרטט במחברתך רק את העץ הפורש (BFS) שמתקבל. התבסס על הייצוג הנתון על ידי רשימת הסמיכויות.



11. בשאלה זו חמישה סעיפים, א-ה. אין קשר בין הסעיפים. עליך לענות על כל הסעיפים.

א. בטבלה שלפניך נתון חלק מפתרון בסיסי אפשרי לבעיית התובלה:  $x_{11} = 100$ ,  $x_{12} = 20$

מקורות	יעדים			היצע
	1	2	3	
1	10 100	15 20	17	120
2	10	18	14	100
3	15	20	18	100
ביקוש	100	110	110	

העתק את הטבלה למחברתך והשלם בה את הפתרון הבסיסי האפשרי לפי שיטת הפינה הצפונית-מערבית.

ב. בטבלה שלפניך נתון חלק מפתרון בסיסי אפשרי לבעיית תובלה, ונתונים ערכיהם של

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

מקורות	יעדים			היצע	$u_i$
	1	2	3		
1	2 20	5	7	20	2
2	1	1 10	4	10	0
3	0	1	8 10	15	0
ביקוש	20	15	10		
$v_j$	0	1	8		

העתק את הטבלה למחברתך, והשלם אותה תוך התחשבות בערכיהם של  $u_i$  ו- $v_j$ , כדי שיתקבל פתרון בסיסי אפשרי.

ג. בטבלה שלפניך נתון פתרון בסיסי אפשרי לבעיית תובלה, ונתונים ערכיהם של  $v_1, v_2, v_3$ ,  $u_1, u_2, u_3$ .

מקורות	יעדים			היצע	$u_i$
	1	2	3		
1	14	15	17	180	0
		130	50		
2	10	8	14	100	-7
	80	20			
3	15	20	18	80	1
			80		
ביקוש	80	150	130		
$v_j$	17	15	17		

האם הפתרון הוא אופטימלי? נמק את תשובתך.

ד. בטבלה שלפניך נתון פתרון בסיסי אפשרי לבעיית תובלה, ונתון  $v_1 = 0$ .

מקורות	יעדים			היצע	$u_i$
	1	2	3		
1	10	25	30	20	
	20				
2	10	22	14	50	
	30		20		
3	18	20	20	60	
		40	20		
ביקוש	50	40	40		
$v_j$	0				

העתק את הטבלה למחברתך והשלם בה את הערכים של  $v_3, v_2, u_3, u_2, u_1$ .

(שים לב: המשך השאלה בעמוד הבא.)

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

מקורות	יעדים			היצע	$u_i$
	1	2	3		
1	20 40	35	26	40	20
2	10 50	22	14 40	90	10
3	28	30 60	30 40	100	26
ביקוש	90	60	80		
$v_j$	0	4	4		

(1) האם הפתרון האופטימלי הוא יחיד?

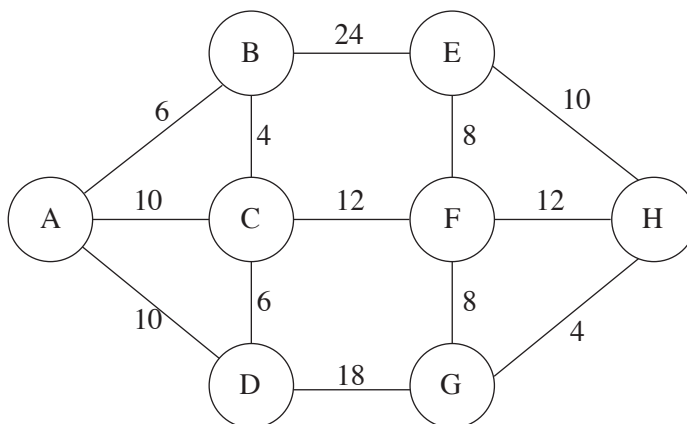
(2) נחסיר 4 מן העלות ליחידה ממקור 3 לכל יעד ונקבל בעיה חדשה.

לפניך ארבעה היגדים ורק אחד מהם נכון. העתק למחברתך את ההיגד הנכון ונמק את בחירתך.

- i הפתרון הנתון איננו פתרון אפשרי בעבור הבעיה החדשה.
- ii הפתרון הנתון הוא פתרון בסיסי אפשרי, אך איננו אופטימלי בעבור הבעיה החדשה.
- iii הפתרון הנתון הוא פתרון אופטימלי יחיד בעבור הבעיה החדשה.
- iv הפתרון הנתון הוא פתרון אופטימלי, אך איננו פתרון אופטימלי יחיד בעבור הבעיה החדשה.

12. בשאלה זו שני סעיפים, א-ב. אין קשר בין הסעיפים. עליך לענות על שניהם.

א. לפניך רשת  $G = (V, E)$  כאשר  $V$  היא קבוצת הקדקודים ברשת ו- $E$  היא קבוצת הקשתות ברשת. פונקציית המשקל  $w: E \rightarrow \mathbb{R}^+$  קובעת משקל (מספר גדול מ-0) לכל קשת ברשת.



מצא את כל המסלולים הקצרים ביותר ברשת הנתונה, החל מקדקוד A לקדקוד H. הצג תיאור סכמתי של כל אחד מן המסלולים שמצאת.

ב. (אין קשר לסעיף א.)

$G = (V, E)$  הוא גרף מכוון. לפניך תיאור חלקי של אלגוריתם הבדוק אם בגרף הנתון יש מעגל.

הפלט של האלגוריתם הוא: קיים מעגל או לא קיים מעגל.

**אלגוריתם:**

**צעד 1:** הפעל את אלגוריתם [1] על הגרף  $G$ .

**צעד 2:** אם [2] מכיל [3], אז

2.1 הפלט הוא: בגרף  $G$  לא קיים מעגל.

2.2 אחרת – הפלט הוא: בגרף  $G$  קיים מעגל.

באלגוריתם חסרים שלושה ביטויים המסומנים [1], [2], [3].

(שים לב: המשך השאלה בעמוד הבא.)

כל אחד מן התת־סעיפים (1)-(3) שלפניך מתייחס לאחד הביטויים החסרים, ומכיל תשובות אפשריות בעבורו.

בכל תת־סעיף נתונות ארבע תשובות, i-iv, ורק אחת מהן נכונה. העתק את האלגוריתם למחברתך, והשלם אותו בעזרת התשובות הנכונות.

(1) התשובה הנכונה עבור ביטוי [1] באלגוריתם היא:

i דיקסטרה

ii סריקה לרוחב (BFS)

iii סריקה לעומק (DFS)

iv מציאת רכיבי קשירות חזקה (רק"חים).

(2) התשובה הנכונה עבור ביטוי [2] באלגוריתם היא:

i המסלול הקצר

ii העץ הפורש (BFS)

iii העץ הפורש (DFS)

iv כל רכיב קשירות חזקה (רק"ח).

(3) התשובה הנכונה עבור ביטוי [3] באלגוריתם היא:

i קדקוד אחד בלבד

ii שני קדקודים לפחות

iii את כל קדקודי הגרף

iv את כל קשתות הגרף

## מודלים חישוביים

אם למדת מסלול זה, ענה על שתיים מן השאלות 13-16 (לכל שאלה – 25 נקודות).

13. בשאלה זו שני סעיפים, א-ב. אין קשר בין הסעיפים. עליך לענות על שניהם.

א. כל מולקולת DNA מורכבת מ-4 אבני בניין יסודיות המסומנות באותיות: A, C, T, G.

אפשר לייצג כל מולקולה כמילה המורכבת מן האותיות A, C, T, G.

במחקר של מולקולת DNA מסוימת נמצא שאם המילה המייצגת את המולקולה מתחילה

ברצף AAG, ובמילה מופיע הרצף AT פעמיים או יותר – הסיכוי למחלה X גבוה.

דוגמה למולקולת DNA שבה הסיכוי למחלה גבוה: המילה המייצגת את

מולקולת ה-DNA: AAGTGAATGATCATGC.

בנה אוטומט סופי דטרמיניסטי שיקבל את השפה של כל המילים המייצגות

מולקולות DNA עם סיכוי גבוה למחלה X.

ב. (אין קשר לסעיף א.)

השפות  $L_2, L_3, L_6$ , מוגדרות מעל הא"ב  $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ .

$L_2$  היא אוסף המילים המייצגות את המספרים המתחלקים ב-2 ללא שארית.

$L_3$  היא אוסף המילים המייצגות את המספרים המתחלקים ב-3 ללא שארית.

$L_6$  היא אוסף המילים המייצגות את המספרים המתחלקים ב-6 ללא שארית.

לפניך שלוש טענות i-iii. לכל אחת מן הטענות קבע אם היא נכונה או אינה נכונה.

אם הטענה נכונה – הסבר מדוע.

אם הטענה אינה נכונה – הסבר מדוע או כתוב מילה המפריכה את הטענה, והסבר מדוע היא

מפריכה אותה.

$$L_6 \subset L_3 \quad i$$

$$L_2 \cap L_3 = L_6 \quad ii$$

$$L_2 \cdot L_3 = L_6 \quad iii$$

14. בכל אחד מן הסעיפים א-ה שלפניך מוגדרת שפה מעל הא"ב  $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ . המילים בכל אחת מן השפות מייצגות מספרים גדולים מ-0.
- בכל אחד מן הסעיפים — בנה אוטומט סופי דטרמיניסטי שיקבל את השפה המוגדרת בו. כל אחד מן האוטומטים יקלוט את המספר ספרה אחרי ספרה, החל בספרת האחדות.
- א. מספרים זוגיים בלבד.
  - ב. מספרים אי-זוגיים קטנים מ-1000.
  - ג. מספרים שמתחלקים ב-5 ללא שארית, וספרת האלפים שלהם, אם קיימת — היא זוגית.
  - ד. מספרים שמתחלקים ב-1000 ללא שארית.
  - ה. מספרים שמתחלקים ב-9 ללא שארית, וקטנים מ-30.

15. בשאלה זו שני סעיפים א-ב. אין קשר בין הסעיפים. ענה על שניהם.

א. נתונה פעולה הכתובה ב־ Java וב־ C#.

Java

```
boolean foo(String str) {
    int cntA = 0;
    int cntC = 0;
    for (int i = 0; i < str.length; i++) {
        if (str[i] == 'a') cntA++;
        if (str[i] == 'c') cntC++;
    }
    if ((cntA % 2 == 0) &&
        (cntC % 3 == 0))
        return true;
    return false;
}
```

C#

```
bool Foo(string str) {
    int cntA = 0;
    int cntC = 0;
    for (int i = 0; i < str.Length; i++) {
        if (str[i] == 'a') cntA++;
        if (str[i] == 'c') cntC++;
    }
    if ((cntA % 2 == 0) &&
        (cntC % 3 == 0))
        return true;
    return false;
}
```

(1) כתוב את השפה  $L$  מעל הא"ב  $\{a, c\}$  שהיא אוסף כל המילים שבעבורן הפעולה

הנתונה מחזירה true.

(2) בנה אוטומט סופי דטרמיניסטי שיקבל את השפה  $L$ .

ב. (אין קשר לסעיף א.)

בנה אוטומט סופי לא דטרמיניסטי מעל הא"ב  $\{a, b\}$  שיקבל את כל המילים המכילות

לפחות מופע אחד של אחד מן הרצפים:  $ababa, aaba, bbb$ .

16. לפניך השפה  $L$  מעל הא"ב  $\{a, b, c\}$ .

$$L = \{a^n b^m c^{n+m} \mid n, m > 0\}$$

א. בנה אוטומט מחסנית שיקבל את השפה  $L$ .

ב. בנה מכונת טיורינג שתקבל את השפה  $L$ .



### תכנות מונחה עצמים

אם למדת מסלול זה ואתה כותב ב־ Java, ענה על שתיים מן השאלות 17-20.  
(לכל שאלה — 25 נקודות)

17. לפניך המחלקות A, B, C, D, E.

```
public class A {
    public void doSomething(int x) {System.out.println("1");}
    public void doSomethingElse(int x, int y) {System.out.println("2");}
    public int calc(int x) {System.out.println("3"); return x + 1;}
}

public class B extends A {
    public void doSomethingElse(int x) {System.out.println("4");}
}

public class C extends A {
    public void doSomething(int x) {System.out.println("5");}
    public int otherCalc(int x) {System.out.println("6"); return x - 1;}
}

public class D extends A {
    public void calc(int x, int y) {System.out.println(7 + y + x);}
}

public class E extends C {
    public int calc(int x) {System.out.println("8"); return x + 3;}
}
```

(שים לב: המשך השאלה בעמוד הבא.)

א. לכל אחת מן ההוראות שלפניך, כתוב אם היא תקינה או אינה תקינה.

אם היא אינה תקינה — נמק מדוע. אם היא תקינה כתוב מה יודפס.

```
A x1 = new A(); x1.doSomethingElse(9,1);
A x2 = new B(); x2.doSomething(9);
A x3 = new B(); x3.doSomethingElse(9,1);
A x4 = new B(); x4.doSomethingElse(9);
A x5 = new C(); x5.doSomething(9);
A x6 = new C(); System.out.println(x6.calc(9));
A x7 = new D(); System.out.println(x7.calc(9));
A x8 = new D(); System.out.println(x8.otherCalc(9));
A x9 = new E(); x9.doSomething(9);
A x10 = new E(); System.out.println(x10.calc(9));
A x11 = new E(); System.out.println(x11.otherCalc(9));
```

ב. בקטע שלפניך מוצגות הוראות ובהן חלקים חסרים, ממוספרים [1]-[5].

```
[1] x = new [2] ();
A y = new [3] ();
x. [4] ([5]);
y.doSomething(11);
```

(i) העתק למחברתך את הספרות [1]-[5], וליד כל אחת מהן כתוב את החסר, כדי שיתקבל הפלט:

99

5

(ii) האם אפשר לכתוב את החסר ב-[3] בצורה שונה מזו שכתבת, כדי שיתקבל הפלט המבוקש? אם אפשר — כתוב זאת, אם אי-אפשר — הסבר מדוע.

18. בשאלה זו עליך לממש מחלקות. בכל פעם שיש לממש מחלקות עליך לכתוב את כותרת המחלקה ואת התכונות שלה, ולממש את כל הפעולות. הנח שהפעולות `get` ו-`set` נתונות ואין צורך לממש אותן.

במסעדה "יאכלו רעבים" משתמשים בתכנה לניהול של מנת הקינוח.

המחלקה **Dessert** מייצגת נתונים של מנת קינוח. במחלקה יש שלוש תכונות:

`name` — מחרוזת, שם מנת הקינוח.

`cost` — מספר שלם, המייצג את העלות של מנה אחת מקינוח זה.

`amount` — מספר שלם, המייצג את מספר הפריטים מקינוח זה.

במחלקה שלוש פעולות:

- פעולה `בונה`, המקבלת שלושה פרמטרים ומאתחלת את התכונות בהתאם.
- פעולה `totalCost`, המחשבת ומחזירה את העלות של כל פריטי הקינוח.
- העלות של כל פריטי הקינוח היא: מספר הפריטים  $\times$  העלות של מנת קינוח אחת.
- פעולה `toString`, המחזירה מחרוזת המורכבת משם מנת הקינוח, מספר הפריטים שיש והעלות של כל פריטי הקינוח.

א. ממש ב-`Java` את המחלקה **Dessert**.

המחלקה **FrozenDessert** יורשת את המחלקה **Dessert**, ומייצגת נתונים של מנות

קינוח קפוא.

למחלקה תכונה נוספת `temperature` — מספר שלם, שמייצג את הטמפרטורה שבה יש לאחסן

את מנת הקינוח.

במחלקה שתי פעולות:

- פעולה `בונה`, המקבלת שלושה פרמטרים (שם, עלות מנת קינוח, מספר הפריטים), ומאתחלת את התכונות בהתאם. הפעולה מאתחלת את התכונה `temperature` ל- $(-4)$ .
- פעולה `toString`, המחזירה מחרוזת המורכבת משם מנת הקינוח, מספר הפריטים, העלות של כל פריטי הקינוח והטמפרטורה שבה יש לאחסן את הקינוח.

ב. ממש ב-`Java` את המחלקה **FrozenDessert**.

(שים לב: המשך השאלה בעמוד הבא.)

המחלקה **Cake** יורשת את המחלקה **Dessert**, ומייצגת נתוני עוגה.

למחלקה תכונה נוספת, `numPieces` – מספר שלם, המייצג את מספר פרוסות העוגה. במחלקה שתי פעולות:

- פעולה בונה, המקבלת ארבעה פרמטרים ומאתחלת את התכונות בהתאם.
  - פעולה `piecePrice`, המחשבת ומחזירה את העלות של פרוסת עוגה אחת.
- שים לב: התכונה `cost` מייצגת עלות של עוגה שלמה.

ג. ממש ב־ Java את המחלקה **Cake**.

ד. ממש ב־ Java מחלקה נוספת בשם **IceCream**, היורשת את המחלקה **FrozenDessert** ומייצגת את מנת הקינוח גלידה.

למחלקה תכונה נוספת: `sprinkles` – משתנה בוליאני שערכו `true` אם הוסיפו לגלידה סוכריות, אחרת – `false`. במחלקה שתי פעולות:

- פעולה בונה, המקבלת ארבעה פרמטרים ומאתחלת את התכונות בהתאם.
- פעולה `toString`, המחזירה מחרוזת המורכבת משם מנת הקינוח, העלות של כל פריטי הקינוח והטמפרטורה שבה יש לאחסן את הקינוח. אם הוסיפו למנה סוכריות – נוספת גם המחרוזת "Sprinkles are free".

ה. כתוב ב־ Java פעולה חיצונית `printDesserts` המקבלת כפרמטר מערך מטיפוס **Dessert** שבתוכו מנות קינוח שונות. על הפעולה להדפיס בעבור כל אחת ממנות הקינוח במערך את עלותה.

בעבור הקינוח עוגה יש להדפיס את עלות העוגה כולה.

19. לפניך הממשקים IOne, ITwo, IThree ו־IFour והמחלקות Alpha, Beta, Gamma,

ו־Program.

```
public interface IOne {
    public boolean firstA(Object x);
    public void firstB(int num);
}
```

```
public interface ITwo {
    public int second ();
}
```

```
public interface IThree {
    public int third();
}
```

```
public interface IFour {
    public int forth();
}
```

```
public class Alpha implements IOne {}
public class Beta implements ITwo, IFour {}
public class Gamma implements ITwo, IThree {}
public class Program {
    public static void main(String[] args) {}
}
```

א. כתוב את שמות הפעולות שיש לממש בכל אחת מן המחלקות Alpha, Beta, ו־Gamma,

והסבר מדוע.

(שים לב: המשך השאלה בעמוד הבא.)

הנח שבכל אחת מן המחלקות מומשו הפעולות שכתבת בסעיף א.

**ב.** האם ההגדרה `public class Omega implements IFour extends Beta { }` תקינה?

אם היא אינה תקינה – הסבר מדוע.

**ג.** לכל אחד מן הקטעים i-iv שלפניך, אם נכתוב אותו בפעולה main – קבע אם הפעולה

main שתקבל היא תקינה או אינה תקינה. נמק את קביעותך.

i `ITwo x1 = new ITwo();`

ii `Beta b = new Beta();`

iii `Alpha a = new Alpha();`

`IOne x2 = a;`

iv `Gamma c = new Gamma ();`

`IOne x3 = c ;`

**ד.** לכל אחת מן הדרישות i-ii שלפניך, קבע אם אפשר לבצע אותה באמצעות כתיבת

הוראה או הוראות בפעולה main. אם אפשר – כתוב את ההוראה המתאימה או את

ההוראות המתאימות. אם אי־אפשר – הסבר מדוע.

i הפעלת הפעולה second על עצם מטיפוס Beta.

ii המרת עצם מטיפוס Gamma שיהיה עצם מטיפוס Alpha.

20. בשאלה זו ארבעה סעיפים, א-ד. אין קשר בין הסעיפים. עליך לענות על כל הסעיפים.

א. נתונות המחלקה A והמחלקה B יורשת מ־ A.

ההוראה: `A a1 = new B();` עוברת הידור (קומפילציה) ורצה בצורה תקינה.

בעבור כל אחד מן המשפטים iii-i שלפניך, קבע אם המשפט נכון או אינו נכון. הסבר את קביעותיך.

i אי־אפשר לכתוב את ההוראה: `Object obj = a1;` מכיוון שדרושה המרה מפורשת (casting).

ii אי־אפשר לכתוב את ההוראה: `A a2 = a1;` מכיוון שדרושה המרה מפורשת (casting).

iii אי־אפשר לכתוב את ההוראה: `B b1 = a1;` מכיוון שדרושה המרה מפורשת (casting).

ב. נתונה הפעולה main במחלקה Program:

```
public static void main(String[] args) {
    C c = new A();
    B b1 = (B) (new A());
    B b2 = new D();
    A a = new D();
}
```

קבע איזו מן האפשרויות v-i שלפניך מתארת קשר בין המחלקות, לפי הפעולה main הנתונה.

i C יורשת מ־ A, B יורשת מ־ A, D יורשת מ־ A.

ii A יורשת מ־ C, B יורשת מ־ A, D יורשת מ־ A.

iii C יורשת מ־ A, B יורשת מ־ A, D יורשת מ־ B.

iv A יורשת מ־ C, C יורשת מ־ B, D יורשת מ־ A.

v A יורשת מ־ C, B יורשת מ־ A, D יורשת מ־ B, C יורשת מ־ B.

(שים לב: המשך השאלה בעמוד הבא.)

ג. נתונות הגדרות של המחלקות School ו־ City :

```
public class City { ____ (1) ____ int students; }
public class School extends City {
    void poll() { System.out.println("There are "+ students + " in the school ") ; }
}
```

כדי שלא תהיה שגיאת הידור (קומפילציה) אפשר להשלים את חלק השורה המסומן (1) באמצעות אחת מן האפשרויות שלפניך. בחר באפשרות המתאימה והעתק אותה למחברתך.

- לכתוב public , private , או protected .
- לכתוב רק public או private .
- לכתוב רק public או protected .
- לכתוב רק protected .
- לכתוב רק public .
- לכתוב רק private .

ד. לפניך הגדרה של המחלקה Quest :

```
public class Quest {
    private int num;
    private static int count = 0;

    public Quest()
    {
        count++;
        num = count;
    }
    public void printNow()
    {
        System.out.println (num + " " + count);
    }
}
```

קבע כמה עצמים (אובייקטים) יש לייצר מן המחלקה Quest , ועל איזה מהם יש להפעיל את הפעולה printNow , כדי שהפלט יהיה 35 .



**תכנות מונחה עצמים**

אם למדת מסלול זה ואתה כותב ב- C#, ענה על שתיים מן השאלות 21-24.

(לכל שאלה — 25 נקודות)

21. לפניך המחלקות A, B, C, D, E.

```
public class A
{
    public virtual void DoSomething(int x) { Console.WriteLine("1"); }
    public void DoSomethingElse(int x, int y) { Console.WriteLine("2"); }
    public virtual int Calc(int x) { Console.WriteLine("3"); return x + 1; }
}

public class B : A
{
    public void DoSomethingElse(int x) { Console.WriteLine("4"); }
}

public class C : A
{
    public override void DoSomething(int x) { Console.WriteLine("5"); }
    public int OtherCalc(int x) { Console.WriteLine("6"); return x - 1; }
}

public class D : A
{
    public void Calc(int x, int y) { Console.WriteLine(7 + y + x); }
}

public class E : C
{
    public override int Calc(int x) { Console.WriteLine("8"); return x + 3; }
}
```

(שים לב: סעיפי השאלה בעמוד הבא.)

א. לכל אחת מן ההוראות שלפניך, כתוב אם היא תקינה או אינה תקינה.  
אם היא אינה תקינה — נמק מדוע, ואם היא תקינה — כתוב מה יודפס.

```
A x1 = new A(); x1.DoSomethingElse(9, 1);
A x2 = new B(); x2.DoSomething(9);
A x3 = new B(); x3.DoSomethingElse(9, 1);
A x4 = new B(); x4.DoSomethingElse(9);
A x5 = new C(); x5.DoSomething(9);
A x6 = new C(); Console.WriteLine(x6.Calc(9));
A x7 = new D(); Console.WriteLine(x7.Calc(9));
A x8 = new D(); Console.WriteLine(x8.OtherCalc(9));
A x9 = new E(); x9.DoSomething(9);
A x10 = new E(); Console.WriteLine(x10.Calc(9));
A x11 = new E(); Console.WriteLine(x11.OtherCalc(9));
```

ב. בקטע שלפניך מוצגות הוראות ובהן חלקים חסרים, ממוספרים [1]-[5].

```
[1] x = new [2] ();
A y = new [3] ();
x. [4] ( [5] );
y.DoSomething(11);
```

(i) העתק למחברתך את הספרות [1]-[5], וליד כל אחת מהן כתוב את החסר

כדי שיתקבל הפלט:

99

5

(ii) האם אפשר להשלים את החסר ב-[3] בצורה שונה מזו שכתבת, כדי שיתקבל הפלט

המבוקש? אם אפשר — כתוב זאת, אם אי-אפשר — הסבר מדוע.

22. בשאלה זו עליך לממש מחלקות. בכל פעם שיש לממש מחלקות עליך לכתוב את כותרת המחלקה ואת התכונות שלה, ולממש את כל הפעולות. הנח שהפעולות Get ו- Set נתונות ואין צורך לממש אותן.

במסעדה "יאכלו רעבים" משתמשים בתכנה לניהול של מנת הקינוח. המחלקה **Dessert** מייצגת נתונים של מנת קינוח. במחלקה יש שלוש תכונות:

name — מחרוזת, שם מנת הקינוח.

cost — מספר שלם, המייצג את העלות של מנה אחת מקינוח זה.

amount — מספר שלם, המייצג את מספר הפריטים מקינוח זה.

במחלקה שלוש פעולות:

- פעולה בונה, המקבלת שלושה פרמטרים ומאתחלת את התכונות בהתאם.
- פעולה **TotalCost**, המחשבת ומחזירה את העלות של כל פריטי הקינוח. העלות של כל פריטי הקינוח היא: מספר הפריטים  $\times$  העלות של מנת קינוח אחת.
- פעולה **ToString**, המחזירה מחרוזת המורכבת משם מנת הקינוח, מספר הפריטים שיש והעלות של כל פריטי הקינוח.

א. ממש ב- C# את המחלקה **Dessert**.

המחלקה **FrozenDessert** יורשת את המחלקה **Dessert**, ומייצגת נתונים של מנות

קינוח קפוא.

למחלקה תכונה נוספת **temperature** — מספר שלם, שמייצג את הטמפרטורה שבה יש לאחסן את מנת הקינוח.

במחלקה שתי פעולות:

- פעולה בונה, המקבלת שלושה פרמטרים (שם, עלות מנת קינוח, מספר הפריטים), ומאתחלת את התכונות בהתאם. הפעולה מאתחלת את התכונה **temperature** ל-  $(-4)$ .
- פעולה **ToString**, המחזירה מחרוזת המורכבת משם מנת הקינוח, מספר הפריטים, העלות של כל פריטי הקינוח והטמפרטורה שבה יש לאחסן את הקינוח.

ב. ממש ב- C# את המחלקה **FrozenDessert**.

(שים לב: המשך השאלה בעמוד הבא.)

המחלקה **Cake** יורשת את המחלקה **Dessert**, ומייצגת נתוני עוגה.

למחלקה תכונה נוספת, `numPieces` – מספר שלם, המייצג את מספר פרוסות העוגה. במחלקה 2 פעולות:

- פעולה בונה, המקבלת ארבעה פרמטרים ומאתחלת את התכונות בהתאם.
  - פעולה `PiecePrice`, המחשבת ומחזירה את העלות של פרוסת עוגה אחת.
- שים לב: התכונה `cost` מייצגת עלות של עוגה שלמה.

ג. ממש ב- `C#` את המחלקה **Cake**.

ד. ממש ב- `C#` מחלקה נוספת בשם **IceCream**, היורשת את המחלקה **FrozenDessert** ומייצגת את מנת הקינוח גלידה.

למחלקה תכונה נוספת: `sprinkles` – משתנה בוליאני שערכו `true` אם הוסיפו לגלידה סוכריות, אחרת – `false`. במחלקה שתי פעולות:

- פעולה בונה, המקבלת ארבעה פרמטרים ומאתחלת את התכונות בהתאם.
- פעולה `ToString`, המחזירה מחרוזת המורכבת משם מנת הקינוח, העלות של כל פריטי הקינוח והטמפרטורה שבה יש לאחסן את הקינוח. אם הוסיפו למנה סוכריות – נוספת גם המחרוזת "Sprinkles are free".

ה. כתוב ב- `C#` פעולה חיצונית `PrintDesserts` המקבלת כפרמטר מערך מטיפוס **Dessert** שבתוכו מנות קינוח שונות. על הפעולה להדפיס בעבור כל אחת ממנות הקינוח במערך את עלותה.

בעבור הקינוח עוגה יש להדפיס את עלות העוגה כולה.

23. לפניך הממשקים IOne, ITwo, IThree ו־IFour והמחלקות Alpha, Beta, Gamma

ו־Program.

```
interface IOne {
    bool FirstA(Object x);
    void FirstB(int num);
}
```

```
interface ITwo {
    int Second();
}
```

```
interface IThree {
    int Third();
}
```

```
interface IFour{
    int Forth();
}
```

```
public class Alpha : IOne {}
public class Beta : ITwo, IFour {}
public class Gamma : ITwo, IThree {}
public class Program {
    public static void Main() {}
}
```

א. כתוב את שמות הפעולות שיש לממש בכל אחת מן המחלקות Alpha, Beta ו־Gamma, והסבר מדוע.

(שים לב: המשך השאלה בעמוד הבא.)

הנח שבכל אחת מן המחלקות מומשו הפעולות שכתבת בסעיף א.

ב. האם ההגדרה `public class Omega : IFour, Beta { }` תקינה?

אם היא אינה תקינה – הסבר מדוע.

ג. לכל אחד מן הקטעים i-iv שלפניך, אם נכתוב אותו בפעולה Main – קבע אם הפעולה

Main שתתקבל היא תקינה או אינה תקינה. נמק את קביעותיך.

i `ITwo x1 = new ITwo();`

ii `Beta b = new Beta();`

iii `Alpha a = new Alpha();`

`IOne x2 = a;`

iv `Gamma c = new Gamma ();`

`IOne x3 = c ;`

ד. לכל אחת מן הדרישות i-ii שלפניך קבע אם אפשר לבצע אותה באמצעות כתיבת

הוראה או הוראות בפעולה Main. אם אפשר – כתוב את ההוראה המתאימה או את

ההוראות המתאימות. אם אי־אפשר – הסבר מדוע.

i הפעלת הפעולה Second על עצם מטיפוס Beta.

ii המרת עצם מטיפוס Gamma שיהיה עצם מטיפוס Alpha.

24. בשאלה זו ארבעה סעיפים, א-ד. אין קשר בין הסעיפים. עליך לענות על כל הסעיפים.

א. נתונות המחלקה A והמחלקה B היורשת מ-A.

ההוראה `A a1 = new B();` עוברת ההידור (קומפילציה) ורצה בצורה תקינה.

בעבור כל אחד מן המשפטים iii-i שלפניך, קבע אם המשפט נכון או אינו נכון. הסבר את קביעותיך.

i אי-אפשר לכתוב את ההוראה `Object obj = a1;` מכיוון שדרושה המרה מפורשת (casting).

ii אי-אפשר לכתוב את ההוראה `A a2 = a1;` מכיוון שדרושה המרה מפורשת (casting).

iii אי-אפשר לכתוב את ההוראה `B b1 = a1;` מכיוון שדרושה המרה מפורשת (casting).

ב. נתונה הפעולה Main במחלקה Program:

```
public static void Main(string[] args) {
    C c = new A();
    B b1 = (B) (new A());
    B b2 = new D();
    A a = new D();
}
```

קבע איזו מן האפשרויות v-i שלפניך מתארת קשר בין המחלקות, לפי הפעולה Main הנתונה.

i C יורשת מ-A, B יורשת מ-A, D יורשת מ-A.

ii A יורשת מ-C, B יורשת מ-A, D יורשת מ-A.

iii C יורשת מ-A, B יורשת מ-A, D יורשת מ-B.

iv A יורשת מ-C, C יורשת מ-B, D יורשת מ-A.

v A יורשת מ-C, B יורשת מ-A, D יורשת מ-B, C יורשת מ-B.

(שים לב: המשך השאלה בעמוד הבא.)

ג. נתונות הגדרות המחלקות City ו- School :

```
public class City { ____ (1) ____ int students; }
public class School : City {
    void Poll() { Console.WriteLine("There are "+ students + " in the school " ); }
}
```

כדי שלא תהיה שגיאת הידור (קומפילציה) אפשר להשלים את חלק השורה המסומן (1) באמצעות אחת מן האפשרויות שלפניך. בחר באפשרות המתאימה והעתק אותה למחברתך.

- לכתוב public , private , או protected .
- לכתוב רק public או private .
- לכתוב רק public או protected .
- לכתוב רק protected .
- לכתוב רק public .
- לכתוב רק private .

ד. לפניך הגדרה של המחלקה Quest :

```
public class Quest {
    private int num;
    private static int count = 0;

    public Quest()
    {
        count++;
        num = count;
    }
    public void PrintNow()
    {
        Console.WriteLine(num+" "+count);
    }
}
```

קבע כמה עצמים (אובייקטים) יש לייצר מן המחלקה Quest , ועל איזה מהם יש להפעיל את הפעולה PrintNow , כדי שהפלט יהיה 35 .

**בהצלחה!**