

סוג הבחינה: א. בגרות לבתי ספר על-יסודיים
ב. בגרות לנבחנים חיצוניים
מועד הבחינה: קיץ תשע"ו, מועד ב
מספר השאלון: 603,899205

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:
- את כל התכניות שאתה נדרש לכתוב בשפת מחשב בפרק הראשון 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 כטייטה (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 רשימה, תור, מחסנית, עץ בינרי וחוליה, בלי לממש אותן. אם אתה משתמש בפעולות נוספות, עליך לממש אותן.

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

1. 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 רשימה **משולשת** באורך 12.

L1: 2 → 5 → 3 → 7 → 2 → 5 → 3 → 7 → 2 → 5 → 3 → 7 → null

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 3/

2. **שים לב:** לשאלה זו שני נוסחים: אחד ב־ Java (עמודים 3-4), ואחד ב־ C# (עמודים 5-6). עבוד על פי השפה שלמדת.

לפותרים 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;
}
```

(שים לב: המשך השאלה בעמוד הבא.)

נתון מערך a :

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

- א. 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: $a[j]$, $a[i]$, j , i , 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 והמספר $k = 10$. 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 והמספר $k = 11$. רשום את הערך המוחזר.
 בטבלת המעקב יש לכלול עמודות בעבור: $a[\text{left}]$, $a[\text{right}]$, 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 ממיון.
- (1) האם טענת היציאה של הפעולה sod תשתנה? נמק את תשובתך.
 (2) האם טענת היציאה של הפעולה what תשתנה? נמק את תשובתך.

/המשך בעמוד 5/

C# – לפותרים ב-

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;
}
```

(שים לב: המשך השאלה בעמוד הבא.)

נתון מערך a:

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

א. 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: $a[i]$, $a[j]$, 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 והמספר $k = 10$. 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 והמספר $k = 11$. 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.

3. א.

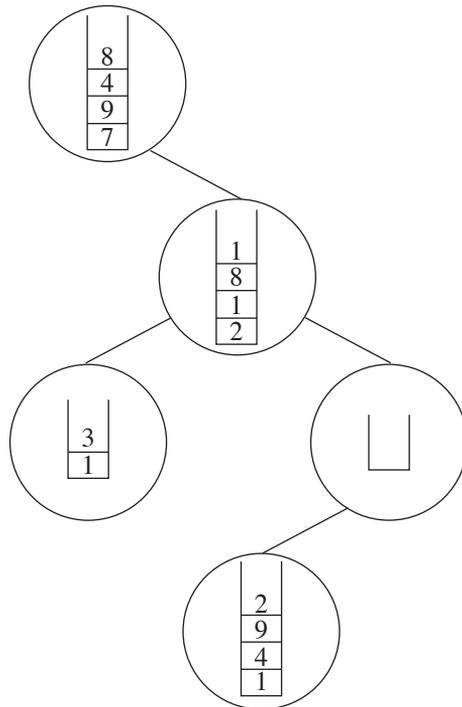
כתוב פעולה חיצונית שתקבל עץ בינארי t לא ריק, שבו כל צומת מכיל מחסנית של מספרים שלמים גדולים מ-0.

הפעולה תחזיר מחסנית. בעבור כל צומת בעץ t יוכנס איבר למחסנית המוחזרת באופן הזה: אם במחסנית שבצומת יש איבר אחד, יוכנס ערכו למחסנית שתוחזר. אם במחסנית שבצומת יש שני איברים, יוכנס סכומם למחסנית שתוחזר. אם המחסנית שבצומת ריקה, יוכנס 0 למחסנית שתוחזר. בכל מקרה אחר יוכנס למחסנית שתוחזר הסכום של שלושת האיברים העליונים של המחסנית שבצומת. סדר האיברים במחסנית שתוחזר יהיה לפי סריקה בסדר תוכי (inorder) של צומתי העץ t . אין צורך לשמור את תוכן המחסניות שבצומתי העץ t . לדוגמה בעבור העץ t :

המחסנית שתוחזר:

0
15
10
4
21

t



ב. מהי סיבוכיות זמן הריצה של הפעולה שכתבת בסעיף א? נמק את תשובתך.

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

.4

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, וציוור 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 9/

- i א. 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, וציור 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. א. 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.

/המשך בעמוד 12/

פרק שני (50 נקודות)

בפרק זה שאלות בארבעה מסלולים:

מערכות מחשב ואסמבלי, עמודים 12-16.

מבוא לחקר ביצועים, עמודים 17-27.

מודלים חישוביים, עמודים 28-29.

תכנות מונחה עצמים ב-Java, עמ' 30-40; תכנות מונחה עצמים ב-C#, עמודים 41-51.

ענה רק על שאלות במסלול שלמדת.

מערכות מחשב ואסמבלי

אם למדת מסלול זה, ענה על שתיים מהשאלות 5-8 (לכל שאלה – 25 נקודות).

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

א. לפניך שיטה לחישוב ריבוע של מספר שלם וגדול מ-0:

$$1^2 = 1$$

$$2^2 = 1 + 3$$

$$3^2 = 1 + 3 + 5$$

$$4^2 = 1 + 3 + 5 + 7$$

$$5^2 = 1 + 3 + 5 + 7 + 9$$

וכן הלאה.

ב- AL מאוחסן מספר שלם, גדול מ-0 וקטן מ-16.

לפניך קטע תכנית באסמבלי שמחשב, לפי השיטה המתוארת, את הריבוע של המספר

המאוחסן ב- AL, ומאחסן את התוצאה ב- AX.

מקטע התכנית הושמטו שורות שלמות או חלקי שורות במקומות המסומנים במספרים

i-iv.

העתק למחברתך את מספרי השורות המסומנות, וכתוב ליד כל אחד מהמספרים את שורת

הפקודה כולל החלקים החסרים, כדי שקטע התכנית יבצע את הנדרש.

		MOV	BX, 0
i		-----	-----
		MOV	DX, 1
		MOV	CX, AX
		DEC	CX
ii		-----	-----
	A1:	ADD	BX, DX
iii		ADD	DX, -----
		LOOP	A1
	A2:	ADD	BX, DX
iv		MOV	AX, -----
	A3:	NOP	

ב. (אין קשר לסעיף א.)

במקטע הנתונים הוגדרו הנתונים כך:

T DB 55 , 90 , 110 , 1

שים לב: המספרים הם עשרוניים.

לפניך קטע תכנית באסמבלי, שמטרתו לחשב את סכום האיברים במערך T.

```
MOV     CX , 4
XOR     AX , AX
LEA     BX , T
AGAIN:  ADD     AL , [BX]
INC     BX
LOOP    AGAIN
```

קטע התכנית אינו מבצע את הנדרש.

(1) עקוב בעזרת טבלת מעקב אחר ביצוע קטע התכנית, וכתוב מה יהיה התוכן של AL, ומה יהיה התוכן של דגל הנשא ושל דגל הגלישה בסיום הביצוע של קטע התכנית.

(2) שנה את קטע התכנית כך שיחשב את סכום האיברים במערך הנתון T. העתק למחברתך את קטע התכנית לאחר השינוי.

/המשך בעמוד 14/

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

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.

i		MOV	SI, 0
		MOV	CX, 4
	A1:	MOV	ARR[SI], 0
		INC	SI
		LOOP	A1
ii		MOV	CX, 5
		LEA	BX, ARR
		MOV	AL, 0
	A1:	MOV	[BX], AL
		INC	BX
	LOOP	A1	
iii		MOV	BX, 5
		DEC	BX
	A1:	MOV	ARR[BX], 0
		DEC	BX
		JNZ	A1
iv		MOV	DI, 0
	A1:	MOV	ARR[DI], 0
		INC	DI
		CMP	DI, 5
		JC	A1

/המשך בעמוד 15/

ב. (אין קשר לסעיף 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 CF, SF, ZF, BX, AX בכל שלב.

/המשך בעמוד 16/

7. במקטע הנתונים מוגדרים שני מערכים. המערך NUMBERS בגודל 100 בתים, המכיל מספרים הקסדצימליים בין 0 ל- OFF, והמערך APP החל מכתובת 2500H, בגודל 256 בתים, המכיל אפסים. כתוב באסמבלי קטע תכנית, שיעבור פעם אחת בלבד על כל איברי המערך NUMBERS, וירשום במקום ה- i במערך APP את מספר הפעמים שהמספר i מופיע במערך NUMBERS.
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.

/המשך בעמוד 17/

מבוא לחקר ביצועים

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

9. נתונה בעיית תכנון לינארי:

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

בכפוף לאילוצים האלה:

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

$$x_1 - x_2 \leq 2$$

$$x_1 \leq 3$$

$$x_2 \geq 0$$

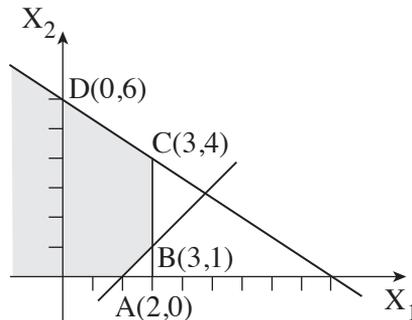
x_1 אינו מוגבל בסימן.

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 כיוון 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 iv-i 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 שישה סעיפים, א-ו, שאינם תלויים זה בזה. ענה על כל הסעיפים.

א. לפניך בעיית תובלה ובה 2 מקורות ו- 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 10	5 2	10	12
2	6	3	6	12
3	6	2	6	6
ביקוש	10	10	10	

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

(שים לב: המשך השאלה בעמוד הבא.)

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					

/המשך בעמוד 21/

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

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

מקורות	יעדים			היצע	u_i
	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.

(שים לב: המשך השאלה בעמוד הבא.)

/המשך בעמוד 22/

ה. בטבלה שלפניך 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.

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

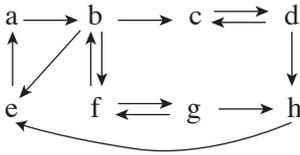
. $X_{21} = 0$ **ii**

. $X_{13} = 0$ **iii**

$X_{31} = 0$ **iv**

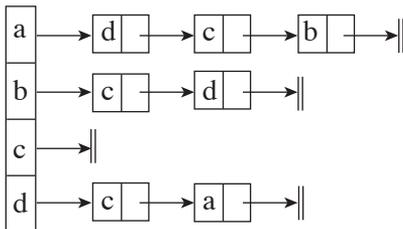
11. בשאלה זו שלושה סעיפים, א-ג, שאינם תלויים זה בזה. ענה על כל הסעיפים.

א. לפניך גרף מכוון G .



- i מצא את רכיב/רכיבי הקשירות החזקה (רק"ח/רק"חיים) שבגרף הנתון.
בעבור כל רק"ח שמצאת, רשום את קבוצת הקדקודים שלו.
- ii האם הגרף הנתון הוא גרף קשיר מכוון בחזקה (קשיר היטב)? נמק את תשובתך.

ב. $G = (V, E)$ הוא גרף מכוון, כאשר V היא קבוצת הקדקודים בגרף ו- E היא קבוצת הקשתות בגרף. G מיוצג על ידי רשימת הסמיכויות שלפניך:



- i הפעל אלגוריתם לחיפוש לעומק (בשיטת DFS) על הגרף הנתון החל מקדקוד a. סרטט במחברתך את העץ הפורש (DFS) / היער הפורש (DFS) שהתקבל.
- ii הפעל אלגוריתם לחיפוש לעומק (בשיטת DFS) על הגרף הנתון החל מקדקוד c. סרטט במחברתך את העץ הפורש (DFS) / היער הפורש (DFS) שהתקבל.
- iii הפעל אלגוריתם לחיפוש לרוחב (בשיטת BFS) על הגרף הנתון החל מקדקוד d. סרטט במחברתך את העץ הפורש (BFS) / היער הפורש (BFS) שהתקבל.

ג. $G = (V, E)$ הוא גרף מכוון, כאשר V היא קבוצת הקדקודים בגרף ו- E היא קבוצת הקשתות בגרף. G מיוצג על ידי רשימת סמיכויות. לפניך תיאור של אלגוריתם אשר בודק אם בגרף G יש מעגל, ומדפיס הודעה מתאימה. באלגוריתם חסרים שני ביטויים המסומנים (1) ו-(2).

תיאור האלגוריתם

צעד 1: הפעל את האלגוריתם למציאת רכיבי קשירות חזקה (SCC) על הגרף G .
(האלגוריתם מניב את הרק"חים הקיימים בגרף.)

צעד 2: אם (1) , אז
הפלט הוא: בגרף G אין מעגל.

צעד 3: אחרת
הפלט הוא: בגרף G יש מעגל.

לפניך שלוש אפשרויות להשלמת ביטוי (1) ושלוש אפשרויות להשלמת ביטוי (2).

האפשרויות:

(1) הוא אחד מאלה:

- אין רק"ח המכיל
- רק"ח אחד לפחות מכיל
- כל אחד מהרק"חים מכיל

(2) הוא אחד מאלה:

- קדקוד אחד בלבד
- שני קדקודים לפחות
- את כל קדקודי הגרף

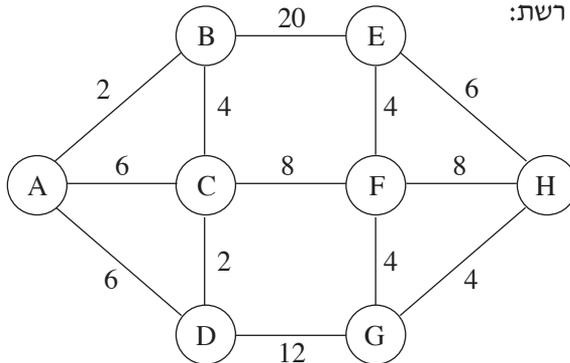
בחר באחת מהאפשרויות להשלמת ביטוי (1) ובאחת מהאפשרויות להשלמת

ביטוי (2), כך שהאלגוריתם יבצע את הנדרש.

כתוב במחברתך את שתי בחירותיך.

12. בשאלה זו שני סעיפים, א-ב, שאינם תלויים זה בזה. ענה על שניהם.

א. לפניך רשת:



רשום במחברתך את חמשת המסלולים הקצרים ביותר מקדקוד A לקדקוד H ברשת הנתונה. תאר כל מסלול כזה בנפרד.

ב. הגרף G הוא קשיר ולא מכוון, ומוגדר על ידי $G = (V, E)$, כאשר V היא קבוצת הקדקודים בגרף ו- E היא קבוצת הקשתות בגרף. פונקציית המשקל $W: E \rightarrow \mathbb{R}^+$ קובעת משקל (מספר) לכל קשת בגרף G.

יהיו X, Y ו- Z קדקודים בגרף G.

נניח כי כל קשת בגרף צבועה בצבע כחול או בצבע אדום. לפניך תיאור של אלגוריתם, אשר בודק אם מבין המסלולים המורכבים מקשתות כחולות בלבד, כל המסלולים הקצרים ביותר מ- X ל- Y עוברים דרך Z. אם כן – האלגוריתם מחזיר את הערך "אמת" (TRUE). אחרת – הוא מחזיר את הערך "שקר" (FALSE). באלגוריתם חסרים 5 ביטויים המסומנים (1)-(5).

תיאור האלגוריתם

צעד 1:

נסיר מהגרף את כל הקשתות (1) ונקבל גרף חדש $G_1 = (V, E_1)$.

צעד 2:

נחשב, בעזרת האלגוריתם של דיקסטר, את אורך המסלול הקצר ביותר מ- X ל- Y בגרף (2). נסמן את אורך המסלול הזה ב- M_1 .

צעד 3:

נסיר מהגרף G_1 את הקדקוד (3) ואת כל הקשתות המחוברות אליו, ונקבל גרף חדש $G_2 = (V_2, E_2)$.

צעד 4:

נחשב, בעזרת האלגוריתם של דיקסטר, את אורך המסלול הקצר ביותר מ- X ל- Y בגרף (4). נסמן את אורך המסלול הזה ב- M_2 .

צעד 5:

אם (5) אזי החזר את הערך "אמת".
אחרת – החזר את הערך "שקר".

העתק למחברתך את הטבלה שלפניך, ורשום בה את הביטויים החסרים.

	ביטוי (1)
	ביטוי (2)
	ביטוי (3)
	ביטוי (4)
	ביטוי (5)

מודלים חישוביים

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

13. נגדיר את השפה L_1 מעל הא"ב $\{a, c\}$:

$$L_1 = \{c^n a^{n+2} \mid n > 0\}$$

לפניך השפה L מעל הא"ב $\{a, b, c\}$:

$$L = \{w_1 w_2 w_3 \dots w_k b^k \mid k > 0, w_i \in L_1\}$$

לדוגמה: המילה caaaccaaaacaaabbb היא מילה בשפה L , כאשר $k = 3$, וכן:

$$w_1 = caaa$$

$$w_2 = ccaaaa$$

$$w_3 = caaa$$

בנה אוטומט מחסנית שיקבל את השפה L .

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

$$L = \left\{ 0^3 \$ 0^{i_1} \$ 0^{i_2} \$ \dots 0^{i_k} \$ \mid \begin{array}{l} k \geq 1 \\ \text{לכל } m \text{ בין } 1 \text{ ל- } k: i_m \geq 0 \\ \text{ו- } i_m \text{ מתחלק ב- } 3 \text{ ללא שארית} \end{array} \right\}$$

א. כתוב את המילה הקצרה ביותר בשפה L .

ב. בנה אוטומט סופי דטרמיניסטי שיקבל את השפה L .

15. לפניך השפה L מעל הא"ב $\{0, 1, 2\}$:

$$L = \{0^n 1^k 2 \mid n > k \geq 0\}$$

א. הוכח כי השפה L אינה רגולרית.

ב. נתונה השפה L_1 מעל הא"ב $\{1, 2\}$:

$$L_1 = \{1^n 2 \mid n \geq 0\}$$

מהי השפה $L \cap L_1$? נמק.

16. 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 רשום 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:

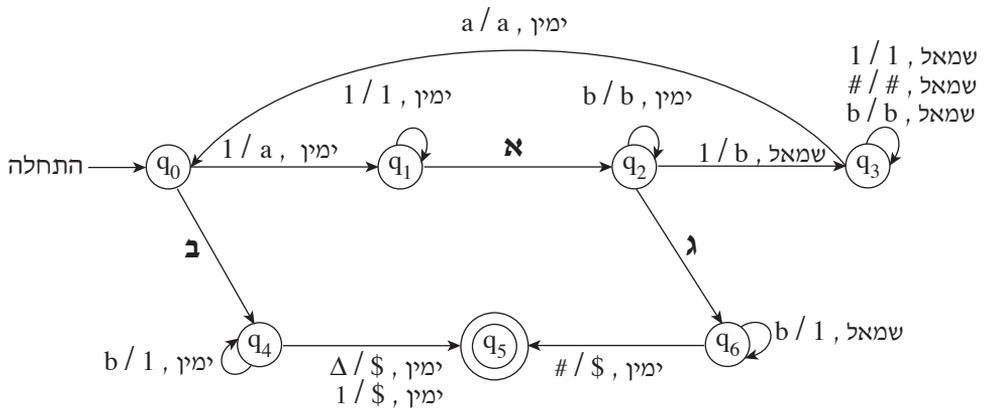
⊢	1	1	1	#	1	1	1	1	1	Δ	Δ	Δ	...
---	---	---	---	---	---	---	---	---	---	---	---	---	-----

לאחר סיום החישוב ייראה הסרט כך:

⊢	\$	1	1	1	\$...
---	-----	-----	-----	----	---	---	---	----	-----

הערה: 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.

העתק 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 $m = 1, n = 1$.

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 שתיים מהשאלות 17-20.
(לכל שאלה – 25 נקודות)

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

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

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 הקרונוות
    public Train(int maxSpeed, int numOfCarriages)
    {
        super("land", "tracks", maxSpeed);
        this.numOfCarriages = numOfCarriages;
    }
    public void incNumOfCarriages(int n) // מגדילה ב־ n את מספר הקרונוות 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.

ג. 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 תקיין 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.

אין 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. אין לשנות את הפעולות הקיימות בפרויקט.

18. 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`.

(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)) 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 משרשור התכונה st של עצמים מטיפוס AA במערך, באופן הזה:

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

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

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

19. חברה המפיצה לומדות פיתחה לומדה העוסקת בסדרות של מספרים שלמים.

המערכת פותחה בשלבים.

בכל סדרה של מספרים מתייחסים אל:

(1) האיבר הראשון בסדרה שמספרו הסידורי הוא 1.

(2) האיבר שמספרו הסידורי בסדרה הוא n .

(3) הדפסת n האיברים הראשונים בסדרה.

בשלב הראשון פותחו שתי מחלקות:

סדרה חשבונית (ASeq) – סדרה שבה ההפרש בין כל איבר לקודמו הוא ערך קבוע.

סדרה הנדסית (GSeq) – סדרה שבה המנה בין כל איבר לקודמו היא ערך קבוע.

להלן קוד המחלקות שפותחו בשלב הראשון:

```
public class ASeq
{
    private int first;
    private int difference;

    public ASeq(int first, int difference)
    {
        this.first = first;
        this.difference = difference;
    }

    public int theNElement(int n)
    {
        return this.first + (n-1) * this.difference;
    }
}
```

(שים לב: המשך השאלה בעמודים הבאים.)

```

public void displayNElements(int n)
{
    System.out.print("The sequence elements: ");
    for (int i = 0; i < n-1; i++)
        System.out.print(this.theNEElement(i+1) + " , ");
    System.out.println(this.theNEElement(n));
}
}

```

```

public class GSeq
{
    private int first;
    private int product;

    public GSeq(int first , int product)
    {
        this.first = first;
        this.product = product;
    }

    public int theNEElement(int n)
    {
        return this.first * (int)Math.pow(this.product , n-1);
    }

    public void displayNElements(int n)
    {
        System.out.print("The sequence elements: ");
        for (int i = 0 ; i < n-1; i++)
            System.out.print(this.theNEElement(i+1) + " , ");
        System.out.println(this.theNEElement(n));
    }
}

```

א. עקוב אחר קטע התכנית שלפניך. במעקב הצג את העצם שנבנה, את התכונות שלו ואת הפלט.

```
ASeq aSeq = new ASeq(2 , 3);
System.out.println(aSeq.theNElement(4));
aSeq.displayNElements(5);
```

ב. **בשלב השני** של הפיתוח הוחלט שמתאים לפתח מחלקה חדשה המתארת סדרה קבועה (Sequence), כך שהמחלקות **ASeq** ו-**GSeq** יירשו מן המחלקה החדשה. בסדרה קבועה מוגדר ערך האיבר הראשון, וכל יתר האיברים זהים לאיבר הראשון.

ג. השלם את הפיתוח של השלב השני באופן המתאים ביותר לעקרונות של תכנות מונחה עצמים, ובהתאם להנחיות (i)-(ii):

(i) ממש באופן מלא את מחלקת העל **Sequence**. המחלקה צריכה להתייחס אל:

(1) האיבר הראשון בסדרה שמספרו הסידורי 1.

(2) האיבר שמספרו הסידורי בסדרה הוא n.

(3) הדפסת n האיברים הראשונים של הסדרה.

(ii) ממש מחדש את המחלקה **ASeq** כך שתירש מן המחלקה **Sequence**.

ד. **בשלב השלישי** של הפיתוח הוחלט להרחיב את הפרויקט שכולל את שלוש המחלקות שפותחו בשלב השני (**Sequence**, **ASeq**, **GSeq**), כך שבעבור כל סדרה יהיה אפשר להפעיל פעולה המחשבת ומחזירה את סכום n האיברים הראשונים של הסדרה.

ה. הנח כי המחלקה **GSeq** מומשה מחדש, כך שהיא יורשת מן המחלקה **Sequence**.

ו. בעבור כל אחת מן המחלקות **Sequence**, **ASeq**, **GSeq**, כתוב אם יש

לעשות בה שינויים כך שהפרויקט יענה על דרישות הפיתוח של השלב השלישי

באופן המתאים ביותר לעקרונות של תכנות מונחה עצמים.

אם יש לעשות שינויים – פרט וממש אותם.

(שים לב: המשך השאלה בעמוד הבא.)

בשלב הרביעי של הפיתוח הוחלט לפתח פעולה סטטית check המקבלת ערך שלם n , ושני עצמים

של סדרות: האחד מטיפוס **ASeq** והאחר מטיפוס **GSeq**.

הפעולה מחשבת את סכום n האיברים הראשונים בכל אחת משתי הסדרות ומחזירה:

— את התו 'A' — אם סכום n האיברים הראשונים של הסדרה מטיפוס **ASeq**
הוא הגדול מבין שני הסכומים.

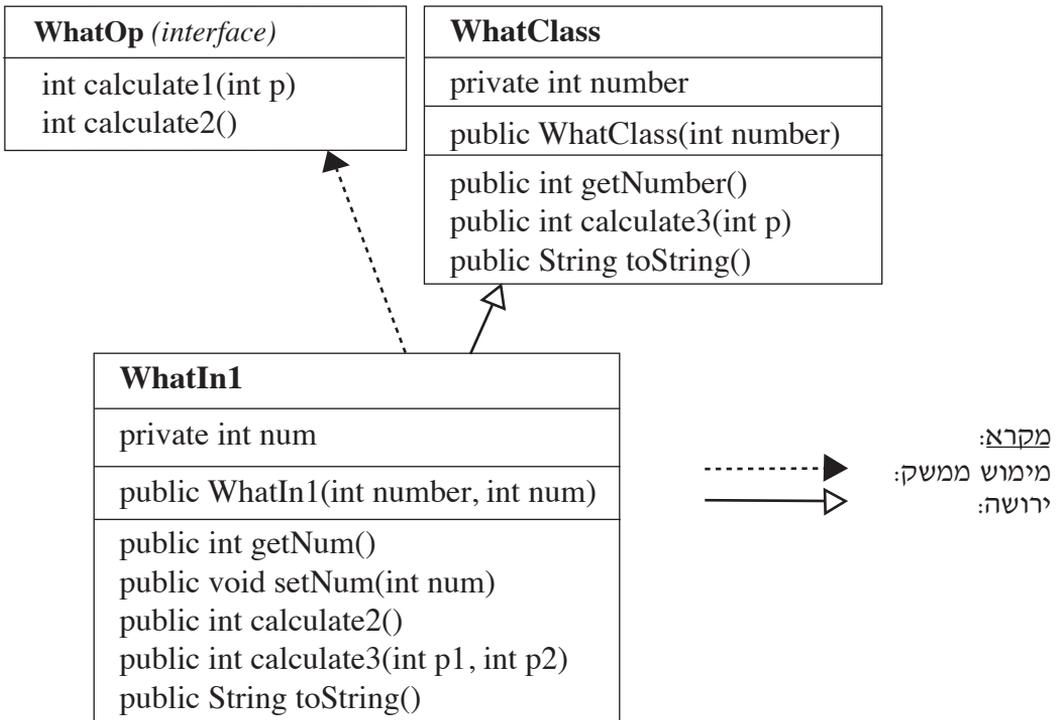
— את התו 'G' — אם סכום n האיברים הראשונים של הסדרה מטיפוס **GSeq**
הוא הגדול מבין שני הסכומים.

— את התו 'E' — אם סכום n האיברים הראשונים של שתי הסדרות שווה.

ד. ממש את הפעולה הסטטית check על פי הדרישות שהוגדרו בשלב הרביעי של הפיתוח.

/המשך בעמוד 39/

20. לפניך UML חלקי:



- א. האם מימוש הפעולה `public int calculate1(int p)` במחלקה **WhatClass**, יענה על דרישת המימוש שלה במחלקה **WhatIn1**? נמק את תשובתך.
- ב. מימוש הפעולה הבונה במחלקה **WhatIn1** הוא:

```

public WhatIn1(int number, int num)
{
    this.number = number;
    this.num = num;
}
    
```

האם הפעולה תקינה? אם כן – תאר את ביצוע הפעולה; אם לא – תקן את הפעולה (אין לשנות דבר פרט לפעולה עצמה). רשום במחברתך את הפעולה המתוקנת.

(שים לב: המשך השאלה בעמוד הבא.)

ג. במחלקה **WhatIn1** מומשה הפעולה `calculate2()`, המחזירה את הערך השלם של ממוצע תכונות העצם:

```
public int calculate2()
{
    return (int)((this.number + this.num)/2);
}
```

האם הפעולה תקינה? אם כן – תאר את ביצוע הפעולה; אם לא – תקן את הפעולה (אין לשנות דבר פרט לפעולה עצמה). רשום במחברתך את הפעולה המתוקנת.

ד. לפניך תיאור חלקי של המחלקה **WhatIn2**:

<code>class WhatIn2 extends WhatIn1</code>
<code>private int sum</code>
<code>public WhatIn2(int number, int num, int sum)</code>
<code>public int calculate3(int p1, int p2, int p3)</code>

(1) האם אפשר להסתמך על הפעולה הבונה ברירת מחדל במקום להגדיר פעולה בונה במחלקה **WhatIn2**? נמק את תשובתך.

(2) הפעולה `calculate3` מומשה בשלוש המחלקות **WhatIn1**, **WhatClass** ו- **WhatIn2** באופן הזה:

<code>class WhatClass</code>	<code>public int calculate3(int p) { return this.number * p; }</code>
<code>class WhatIn1</code>	<code>public int calculate3(int p1, int p2) { return this.calculate3(p1) + this.num * p2 * p2; }</code>
<code>class WhatIn2</code>	<code>public int calculate3(int p1, int p2, int p3) { return this.calculate3(p1, p2) + this.sum * p3 * p3 * p3; }</code>

הנח כי העצם `obj` הוא מטיפוס **WhatIn2**, וערכי התכונות שלו הם:

`number = 1`

`num = 2`

`sum = 3`

לפניך הוראה הכתובה בפעולה הראשית:

```
System.out.println(obj.calculate3(1000, 100, 10));
```

הראה את המעקב אחר ביצוע ההוראה. במעקב התייחס לזימון פעולות ולערכי

תכונות העצם. רשום את הפלט המתקבל.

תכנות מונחה עצמים

אם למדת מסלול זה ואתה כותב ב- **C#**, ענה על שתיים מהשאלות 21-24.
(לכל שאלה – 25 נקודות)

21. 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 הקרונוות

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

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 לחברה 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

אין לשנות את הפעולה Display() .

ג. ממש ב- C# 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. אין לשנות את הפעולות הקיימות בפרויקט.

22. 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 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 `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)) 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 משרשור התכונה st של עצמים מטיפוס AA

במערך, באופן הזה:

— אם לעצם יש ב התכונה st, תשורשר המחרוזת שבתכונה st פעם אחת.

— אם לעצם יש ג התכונה num, המחרוזת שבתכונה st תשורשר num פעמים.

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

23. חברה המפיצה לומדות פיתחה לומדה העוסקת בסדרות של מספרים שלמים. המערכת פותחה בשלבים.

עבור כל סדרה של מספרים מתייחסים אל:

(1) האיבר הראשון בסדרה שמספרו הסידורי הוא 1.

(2) האיבר שמספרו הסידורי בסדרה הוא n .

(3) הדפסת n האיברים הראשונים בסדרה.

בשלב הראשון פותחו שתי המחלקות:

סדרה חשבונית (**ASeq**) – סדרה שבה ההפרש בין כל איבר לקודמו הוא ערך קבוע.

סדרה הנדסית (**GSeq**) – סדרה שבה המנה בין כל איבר לקודמו היא ערך קבוע.

להלן קוד המחלקות שפותחו בשלב הראשון:

```
public class ASeq
{
    private int first;
    private int difference;

    public ASeq(int first, int difference)
    {
        this.first = first;
        this.difference = difference;
    }

    public int TheNElement(int n)
    {
        return this.first + (n-1) * this.difference;
    }
}
```

```

public void DisplayNElements(int n)
{
    Console.WriteLine("The sequence elements: ");
    for (int i = 0; i < n-1; i++)
        Console.WriteLine(this.TheNElement(i+1) + " , ");
    Console.WriteLine(this.TheNElement(n));
}
}

public class GSeq
{
    private int first;
    private int product;

    public GSeq(int first, int product)
    {
        this.first = first;
        this.product = product;
    }

    public int TheNElement(int n)
    {
        return this.first * (int)Math.Pow(this.product , n-1);
    }

    public void DisplayNElements(int n)
    {
        Console.WriteLine("The sequence elements: ");
        for (int i = 0; i < n-1; i++)
            Console.WriteLine(this.TheNElement(i+1) + " , ");
        Console.WriteLine (this.TheNElement(n));
    }
}

```

(שים לב: המשך השאלה בעמודים הבאים.)

א. עקוב אחר קטע התכנית שלפניך. במעקב הצג את העצם שנבנה, את התכונות שלו ואת הפלט.

```
ASeq aSeq = new ASeq(2 , 3);  
Console.WriteLine(aSeq.TheNElement(4));  
aSeq.DisplayNElements(5);
```

בשלב השני של הפיתוח הוחלט שמתאים לפתח מחלקה חדשה המתארת סדרה קבועה (Sequence), כך שהמחלקות **ASeq** ו-**GSeq** יירשו מן המחלקה החדשה. בסדרה קבועה מוגדר ערך האיבר הראשון, וכל יתר האיברים זהים לאיבר הראשון.

ב. השלם את הפיתוח של השלב השני באופן המתאים ביותר לעקרונות של תכנות מונחה עצמים ובהתאם להנחיות (i)-(ii):

(i) ממש באופן מלא את מחלקת העל **Sequence**. המחלקה צריכה להתייחס אל:

(1) האיבר הראשון בסדרה שמספרו הסידורי 1.

(2) האיבר שמספרו הסידורי בסדרה הוא n.

(3) הדפסת n האיברים הראשונים של הסדרה.

(ii) ממש מחדש את המחלקה **ASeq** כך שתירש מן המחלקה **Sequence**.

בשלב השלישי של הפיתוח הוחלט להרחיב את הפרויקט שכולל את שלוש המחלקות שפותחו בשלב השני (**Sequence**, **ASeq**, **GSeq**), כך שבעבור כל סדרה יהיה אפשר להפעיל פעולה המחשבת ומחזירה את סכום n האיברים הראשונים של הסדרה.

הנח כי המחלקה **GSeq** מומשה מחדש כך שהיא יורשת מן המחלקה **Sequence**.

ג. בעבור כל אחת מן המחלקות **GSeq**, **ASeq**, **Sequence**, כתוב אם

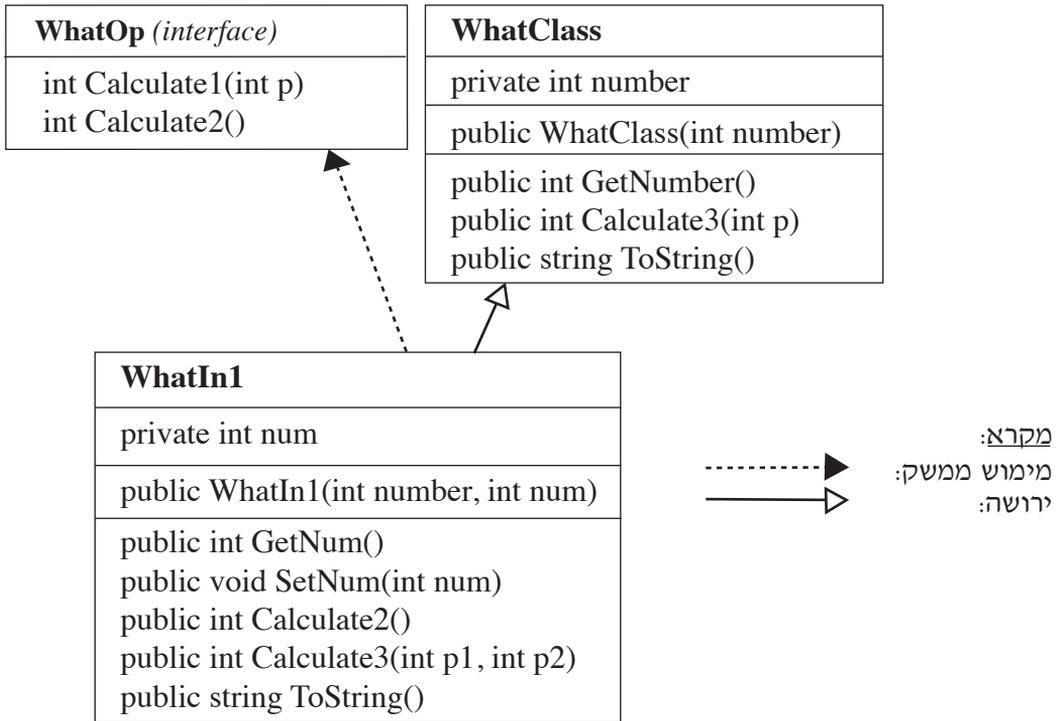
יש לעשות בה שינויים כך שהפרויקט יענה על דרישות הפיתוח של השלב השלישי

באופן המתאים ביותר לעקרונות של תכנות מונחה עצמים.

אם יש לעשות שינויים – פרט וממש אותם.

- בשלב הרביעי** של הפיתוח הוחלט לפתח פעולה סטטית Check המקבלת ערך שלם n , ושני עצמים של סדרות: האחד מטיפוס **ASeq** והאחר מטיפוס **GSeq**.
- הפעולה מחשבת את סכום n האיברים הראשונים בכל אחת משתי הסדרות ומחזירה:
- את התו 'A' — אם סכום n האיברים הראשונים של הסדרה מטיפוס **ASeq** הוא הגדול מבין שני הסכומים.
 - את התו 'G' — אם סכום n האיברים הראשונים של הסדרה מטיפוס **GSeq** הוא הגדול מבין שני הסכומים.
 - את התו 'E' — אם סכום n האיברים הראשונים של שתי הסדרות שווה.
- ד.** ממש את הפעולה הסטטית Check על פי הדרישות שהוגדרו בשלב הרביעי של הפיתוח.

/המשך בעמוד 50/



- א. האם מימוש הפעולה `public int Calculate1(int p)` במחלקה **WhatClass**, יענה על דרישת המימוש שלה במחלקה **WhatIn1**? נמק את תשובתך.
- ב. מימוש הפעולה הבונה במחלקה **WhatIn1** הוא:

```

public WhatIn1(int number, int num)
{
    this.number = number;
    this.num = num;
}
    
```

האם הפעולה תקינה? אם כן – תאר את ביצוע הפעולה; אם לא – תקן את הפעולה (אין לשנות דבר פרט לפעולה עצמה). רשום במחברתך את הפעולה המתוקנת.

ג. במחלקה **WhatIn1** מומשה הפעולה `Calculate2()`, המחזירה את הערך השלם של ממוצע תכונות העצם:

```
public int Calculate2()
{
    return (int)((this.number + this.num)/2);
}
```

האם הפעולה תקינה? אם כן – תאר את ביצוע הפעולה; אם לא – תקן את הפעולה (אין לשנות דבר פרט לפעולה עצמה). רשום במחברתך את הפעולה המתוקנת.

ד. לפניך תיאור חלקי של המחלקה **WhatIn2**:

<code>class WhatIn2: WhatIn1</code>
<code>private int sum</code>
<code>public WhatIn2(int number, int num, int sum)</code>
<code>public int Calculate3(int p1, int p2, int p3)</code>

(1) האם אפשר להסתמך על הפעולה הבונה בִּרְת מחדל במקום להגדיר פעולה בונה במחלקה **WhatIn2**? נמק את תשובתך.

(2) הפעולה `Calculate3` מומשה בשלוש המחלקות **WhatIn1**, **WhatClass** ו־ **WhatIn2** באופן הזה:

<code>class WhatClass</code>	<code>public int Calculate3(int p) { return this.number * p; }</code>
<code>class WhatIn1</code>	<code>public int Calculate3 (int p1, int p2) { return this.Calculate3(p1) + this.num * p2 * p2; }</code>
<code>class WhatIn2</code>	<code>public int Calculate3 (int p1, int p2, int p3) { return this.Calculate3(p1, p2) + this.sum * p3 * p3 * p3;} </code>

הנח כי העצם `obj` הוא מטיפוס **WhatIn2**, וערכי התכונות שלו הם:

`number = 1`

`num = 2`

`sum = 3`

לפניך הוראה הכתובה בפעולה הראשית:

```
Console.WriteLine(obj.Calculate3(1000, 100, 10));
```

הראה את המעקב אחר ביצוע ההוראה. במעקב התייחס לזימון פעולות ולערכי תכונות העצם. רשום את הפלט המתקבל.

בהצלחה!